[MS-AXL2]:
Access Application Transfer Protocol Structure Version 2

Revision Summary

Date	Revision History	Revision Class	Comments
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.0	Major	Significantly changed the technical content.
02/11/2013	2.0	Major	Significantly changed the technical content.
07/30/2013	2.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction 8
1.1 Glossary 8
1.2 References 9
1.2.1 Normative References 9
1.2.2 Informative References 9
1.3 Overview 9
1.3.1 Tables 10
1.3.1.1 Check Constraints 10
1.3.1.2 Default Constraints 10
1.3.1.3 Indexes 10
1.3.1.4 Unique Constraints 10
1.3.1.5 Annotation Elements and Attributes 10
1.3.2 UI Macros 10
1.3.3 Data Macros 11
1.3.4 Queries 11
1.4 Relationship to Protocols and Other Structures 12
1.5 Applicability Statement 12
1.6 Versioning and Localization 12
1.7 Vendor-Extensible Fields 12

2 Structures 13
2.1 Conceptual Overview 13
2.1.1 Tables 13
2.1.1.1 Check Constraints 13
2.1.1.2 Default Constraints 13
2.1.1.3 Indexes 13
2.1.1.4 Unique Constraints 13
2.1.1.5 Annotation Elements and Attributes 14
2.1.1.6 Linked Tables 15
2.1.2 Macros 15
2.1.2.1 User Interface Macros 16
2.1.2.2 Data Macros 16
2.1.2.3 Variables 16
2.1.2.3.1 User Interface Macro Variables 16
2.1.2.3.2 Local Variables 16
2.1.2.3.3 Return Variables 16
2.1.3 Queries 16
2.1.3.1 Aggregate Expression 17
2.1.4 Ignored Value 17
2.2 AXL2 18
2.2.1 Elements 18
2.2.1.1 Query 18
2.2.1.2 Ordering 18
2.2.1.3 UserInterfaceMacros 18
2.2.1.4 DataMacros 18
2.2.1.5 UserInterfaceMacro 19
2.2.1.6 CheckConstraint 19
2.2.1.7 DefaultConstraint 19
2.2.1.8 Index 20
2.2.1.9 Unique 20

2.2.1.10 Expression 20
2.2.1.11 EventDataMacro 20
2.2.1.12 FormDef 21
2.2.1.13 ExtendedAttributes 21
2.2.1.14 DataMacro 21
2.2.1.15 PropertyRef 21
2.2.2 Attributes 22
2.2.2.1 SiteURL 22
2.2.2.2 InternalName 22
2.2.2.3 TextType 22
2.2.2.4 UnderlyingType 22
2.2.2.5 BoundColumn 23
2.2.2.6 Caption 23
2.2.2.7 CurrencySymbol 23
2.2.2.8 Description 24
2.2.2.9 Direction 24
2.2.2.10 DisplayColumn 24
2.2.2.11 DisplayName 24
2.2.2.12 DisplayOrder 25
2.2.2.13 Format 25
2.2.2.14 LookupType 25
2.2.2.15 ObjectId 25
2.2.2.16 ReferencedTable 26
2.2.2.17 StoreGeneratedPattern 26
2.2.2.18 Values 26
2.2.2.19 CurrencyLocale 26
2.2.2.20 DecimalPlaces 27
2.2.3 Complex Types 27
2.2.3.1 CT_Expression 27
2.2.3.2 CT_Order 27
2.2.3.3 CT_Ordering 28
2.2.3.4 CT_AdHocOrder 29
2.2.3.5 CT_AdHocOrdering 29
2.2.3.6 CT_Reference 30
2.2.3.7 CT_References 31
2.2.3.8 CT_Result 31
2.2.3.9 CT_Results 32
2.2.3.10 CT_Join 33
2.2.3.11 CT_Joins 34
2.2.3.12 CT_Parameter 34
2.2.3.13 CT_Parameters 34
2.2.3.14 CT_ParameterDefinition 35
2.2.3.15 CT_ParameterDefinitions 35
2.2.3.16 CT_ParameterValue 36
2.2.3.17 CT_ParameterValues 36
2.2.3.18 CT_Query 37
2.2.3.19 CT_Argument 38
2.2.3.20 CT_Action 39
2.2.3.21 CT_ForEachRecordData 40
2.2.3.22 CT_ForEachRecord 40
2.2.3.23 CT_LookupRecordData 41
2.2.3.24 CT_LookupRecord 42
2.2.3.25 CT_CreateRecordData 42

2.2.3.26 CT_CreateRecord 43
2.2.3.27 CT_EditRecordData 44
2.2.3.28 CT_EditRecord 44
2.2.3.29 CT_UserInterfaceIf 45
2.2.3.30 CT_UserInterfaceElseIf 45
2.2.3.31 CT_UserInterfaceElse 46
2.2.3.32 CT_DataIf 47
2.2.3.33 CT_DataElseIf 47
2.2.3.34 CT_DataElse 48
2.2.3.35 CT_UserInterfaceConditionalBlock 48
2.2.3.36 CT_DataConditionalBlock 49
2.2.3.37 CT_UserInterfaceMacroStatements 50
2.2.3.38 CT_DataMacroStatements 51
2.2.3.39 CT_UserInterfaceMacroStatementGroup 52
2.2.3.40 CT_DataMacroStatementGroup 52
2.2.3.41 CT_UserInterfaceMacros 53
2.2.3.42 CT_DataMacros 53
2.2.3.43 CT_UserInterfaceMacro 54
2.2.3.44 CT_DataMacro 54
2.2.3.45 CT_ExpressionContents 55
2.2.3.46 CT_FunctionCall 56
2.2.3.47 CT_Identifier 60
2.2.3.48 CT_IntegerLiteral 60
2.2.3.49 CT_NullLiteral 61
2.2.3.50 CT_StringLiteral 61
2.2.3.51 CT_DateTimeLiteral 61
2.2.3.52 CT_TimeLiteral 62
2.2.3.53 CT_DateLiteral 62
2.2.3.54 CT_BitLiteral 63
2.2.3.55 CT_IndexedFunctionCall 63
2.2.3.56 CT_IndexedIdentifier 66
2.2.3.57 CT_IndexedIntegerLiteral 67
2.2.3.58 CT_IndexedNullLiteral 67
2.2.3.59 CT_IndexedStringLiteral 68
2.2.3.60 CT_IndexedDatePartLiteral 68
2.2.3.61 CT_IndexedTypeLiteral 69
2.2.3.62 CT_IndexedDateTimeLiteral 69
2.2.3.63 CT_IndexedTimeLiteral 70
2.2.3.64 CT_IndexedDateLiteral 70
2.2.3.65 CT_IndexedBitLiteral 71
2.2.3.66 CT_Index 71
2.2.3.67 CT_Unique 72
2.2.3.68 CT_ConstraintWithExpression 73
2.2.3.69 CT_CheckConstraint 73
2.2.3.70 CT_DefaultConstraint 74
2.2.3.71 CT_OrderExpression 75
2.2.3.72 CT_Group 75
2.2.3.73 CT_GroupExpression 76
2.2.3.74 CT_Groups 76
2.2.3.75 CT_OutputParameter 77
2.2.3.76 CT_TopRows 78
2.2.3.77 CT_TopPercent 78
2.2.3.78 CT_ExpressionArgument 78

2.2.3.79 CT_Expressions 79
2.2.3.80 CT_EventDataMacro 79
2.2.3.81 CT_FormDef 80
2.2.3.82 CT_ExtendedAttribute 81
2.2.3.83 CT_ExtendedAttributes 81
2.2.3.84 CT_DecimalLiteral 82
2.2.3.85 CT_IndexedDecimalLiteral 82
2.2.3.86 CT_NamedExpression 83
2.2.3.87 CT_PropertyRef 84
2.2.3.88 CT_PropertyRefIndex 84
2.2.4 Simple Types 85
2.2.4.1 ST_ObjectName 85
2.2.4.2 ST_DataObjectEvent 85
2.2.4.3 ST_ShortString 86
2.2.4.4 ST_LongString 86
2.2.4.5 ST_JoinDirection 87
2.2.4.6 ST_FieldType 87
2.2.4.7 ST_SortDirection 88
2.2.4.8 ST_TextType 89
2.2.4.9 TStoreGeneratedPattern 89
2.2.4.10 ST_LookupType 90
2.2.4.11 ST_Percent 90
2.2.4.12 ST_UserInterfaceObjectEvent 91
2.2.4.13 ST_Format 92
2.2.4.14 ST_Decimal 93
2.2.4.15 ST_DatePartLiteral 93
2.2.4.16 ST_TypeLiteral 94
2.2.4.17 ST_DecimalPlaces 95
2.2.4.18 ST_QuerySourceType 95
2.2.5 Macros 96
2.2.5.1 Actions 96
2.2.5.1.1 ChangeView 96
2.2.5.1.2 CancelRecordChange 97
2.2.5.1.3 ClosePopup 97
2.2.5.1.4 DeleteRecord 97
2.2.5.1.5 EditRecord 97
2.2.5.1.6 ExitForEachRecord 97
2.2.5.1.7 GoToControl 97
2.2.5.1.8 GoToRecord 98
2.2.5.1.9 MessageBox 98
2.2.5.1.10 OpenPopup 98
2.2.5.1.11 RaiseError 98
2.2.5.1.12 RequeryRecords 99
2.2.5.1.13 RunDataMacro 99
2.2.5.1.14 RunMacro 99
2.2.5.1.15 SaveRecord 99
2.2.5.1.16 SetField 100
2.2.5.1.17 SetLocalVar 100
2.2.5.1.18 SetProperty 100
2.2.5.1.19 SetReturnVar 101
2.2.5.1.20 SetVariable 101
2.2.5.1.21 StopMacro 101
2.2.5.2 Arguments 101

2.2.5.2.1 Alias 101
2.2.5.2.2 ControlName 101
2.2.5.2.3 Description 102
2.2.5.2.4 Field 102
2.2.5.2.5 Table 102
2.2.5.2.6 View 102
2.2.5.2.7 MacroName 102
2.2.5.2.8 Message 102
2.2.5.2.9 Name 102
2.2.5.2.10 Variable 102
2.2.5.2.11 OrderBy 103
2.2.5.2.12 Property 103
2.2.5.2.13 Record 103
2.2.5.2.14 Value 103
2.2.5.2.15 Where 104

3 Structure Examples 105
3.1 Tables 105
3.2 Forms 107
3.3 UI Macros 108
3.3.1 Validation and Exceptions 108
3.3.2 Standalone 109
3.4 Data Macros 109
3.4.1 Validate Delete 109
3.4.2 After Insert 110
3.4.3 Named Macro 111
3.5 Queries 113
3.5.1 Single Reference Query 113
3.5.2 Multiple Reference Query with Join 114

4 Security 116
4.1 Security Considerations for Implementers 116
4.2 Index of Security Fields 116

5 Appendix A: Full XML Schema 117
5.1 http://schemas.microsoft.com/office/accessservices/2010/12/application Schema 117

6 Appendix B: Product Behavior 132

7 Change Tracking 133

8 Index 134

1 Introduction

The Access Application Transfer Protocol Structure Version 2 specification, a revision of [MS-AXL], specifies structures that can be used to describe data application components. This version includes structures for layout descriptions of data views; workflow control; combining, filtering and ordering of data; and metadata describing the entire database application.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

XML

The following terms are defined in [MS-OFCGLOS]:

alias
Boolean
calculated column
caption
column
control
data macro
data source
data type
database application
database object
display name
expression
field
filter
floating-point number
form
hyperlink
Hypertext Markup Language (HTML)
input source
lookup field
query
record
row
session
site
sort order
Uniform Resource Locator (URL)

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MC-CSDL] Microsoft Corporation, "Conceptual Schema Definition File Format".

[MSDN-TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference", http://msdn.microsoft.com/en-us/library/ms189826(SQL.90).aspx

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, August 2006, http://www.w3.org/TR/2006/REC-xml-20060816/

[XMLSCHEMA1] Thompson, H.S., Beech, D., Maloney, M., Eds., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

[MS-AADT] Microsoft Corporation, "Access Application Design Time Protocol".

[MS-ADR] Microsoft Corporation, "Access Services Data Run Time Protocol".

[MS-AXL] Microsoft Corporation, "Access Application Transfer Protocol Structure Specification".

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

[MS-SSDL] Microsoft Corporation, "Store Schema Definition Language File Format Structure Specification", http://msdn.microsoft.com/en-us/library/ff952819.aspx

1.3 Overview

The structures specified by this protocol can be used to define objects in a **database application** such as constraints and indexes on a table, application variables, user interface command routines, data manipulation routines, and **queries**.

This protocol follows XML formatting and does not specify how to contain data or data schemas.

1.3.1 Tables

A table in a database application is a data structure that organizes data as **rows (1)** and **columns (1)**. This protocol defines structures and annotation attributes to describe constraints, indexes and other properties on a table.

This protocol does not support describing certain properties of a table schema, such as columns and relationships. However, this protocol can be used in conjunction with other protocols such as the Conceptual Schema Definition File Format ([MC-CSDL]) or the Store Schema Definition Language File Format ([MS-SSDL]) for describing a complete table schema.

1.3.1.1 Check Constraints

A check constraint is a validation condition for data being added or updated in a table. This protocol allows check constraints to be specified on a table or on a single column. This protocol specifies check constraints by using the **CheckConstraint** element (section 2.2.1.6).

1.3.1.2 Default Constraints

A default constraint on a column (1) specifies a value that the column will take when no value is explicitly assigned to it during the addition of a data row (1). This protocol specifies default constraints by using the **DefaultConstraint** element (section 2.2.1.7).

1.3.1.3 Indexes

An index defined on one or more columns (1) provides faster retrieval of data based on the values within those columns. This protocol specifies indexes by using the **Index** element (section 2.2.1.8).

1.3.1.4 Unique Constraints

A unique constraint on a column (1) enforces uniqueness of its data by not allowing duplicate values in the column. This protocol specifies unique constraints by using the **Unique** element (section 2.2.1.9).

1.3.1.5 Annotation Elements and Attributes

This protocol specifies several annotation element and attribute types (section 2.1.1.5) that can be used with other protocols, such as the Conceptual Schema Definition File Format ([MC-CSDL]), to describe properties of a table schema.

The [MS-AADT] protocol, for example, uses the annotation elements and attributes in the [MC-CSDL] protocol to define a table schema, as described by section 2.1.1.5.

1.3.2 UI Macros

User Interface (UI) Macros are command routines that respond to user interaction and change the state of the UI. They can be used to perform a variety of actions, including navigating among forms, validating and verifying user input, and filtering records that are shown in the forms. Important features of UI Macros include **expressions**, **filters**, and the ability to implement conditional logic. A UI Macro can exist as a standalone object with a name or as an embedded macro that is part of a **form (1)**. Macros that are part of a form are triggered by events that occur on the form or on a control on the form.

An example of application logic is the use of UI Macros to respond to user input in a search field. After the user modifies the contents of a search box and navigates to a different control, a UI Macro

can be triggered that applies a filter to a list of items, such as a list box or a subform, which limits the records displayed in the list to those containing the search term.

Another example is the use of UI Macros to implement navigation functionality, such as drill-through behavior on a form. After an item in a list box is double-clicked, a UI Macro can be triggered that launches a new form, which displays details of the selected item, such as information from a related **data source (1)**.

This protocol specifies UI Macros using the **CT_UserInterfaceMacro** type (section 2.2.3.43). Both embedded and standalone UI Macros can specify a **Statements** element, which specifies commands that will be performed when the Macro is run. Embedded Macros also specify the **Event** attribute, which specifies when the Macro will be triggered, and the optional **For** attribute if the Macro will be triggered by the event of a control on the form, as opposed to an event of the form itself.

1.3.3 Data Macros

Data Macros represent data validation logic, product logic and business logic in a data application. Data Macros allow a data application to perform a series of actions based on a given set of conditions. They can exist as a standalone named object, or be associated with an event in a Table (section 1.3.1).

An example of validation logic is the use of a Data Macro (section 2.1.2.2) to ensure that a value entered into a column (1) makes sense given other values in the Table (section 1.3.1). After a value is entered into the **EndDate** field of a **record**, a Data Macro (section 2.1.2.2) can be triggered to ensure that the value of that field is later than the value of the **StartDate** field in the same record, or else the record will not be updated.

Another example is the use of a Data Macro (section 2.1.2.2) to keep the value of a field in one Table (section 1.3.1) updated based on entries in another Table (section 1.3.1). When a new item is created in a Projects table, a Data Macro (section 2.1.2.2) can be triggered to increment the **TotalProjectCount** field in the separate Managers table, so that each record in the Managers table always has an up-to-date count of the total number of projects assigned to that manager.

Data Macros (section 2.1.2.2) in this protocol are specified by the **CT_DataMacro** type (section 2.2.3.44). All Data Macros (section 2.1.2.2) can specify a **Statements** child element which specifies the commands that will be performed when the Macro is called. Data Macro (section 2.1.2.2) associated with a Table (section 1.3.1) event specify the **Event** attribute, which specifies when the Macro will be triggered. Named Data Macro (section 2.1.2.2) can specify a **Parameters** child element which specifies the names and data types of passed-in parameters.

1.3.4 Queries

Queries are used in a database application for combining, filtering, sorting and grouping of data. Queries can reduce the load on a system by returning only the subset of information that is relevant to a process or an end user.

This structure specifies queries using the **Query** (section 2.2.1.1) element. Queries specified in this structure are similar to SQL SELECT statements and support:

- Projection: Limiting the results to a subset of the columns (1) in the underlying data source (1).
- Ordering: Sorting the results on a combination of one or more columns (1) and expressions.
- Restriction: Limiting the results to certain records, based on data in the record.
- Joins: Creating a new result set from the combination of multiple data sources (1).

- Top Results: Limiting the results to return the first specified number or percentage of records.
- Distinct Results: Limiting the results to return unique rows (1).
- Grouping: Describing the groups into which records will be placed in the result set.
- Group Restriction: Limiting the results to certain groups of records, based on data in the group of records.

1.4 Relationship to Protocols and Other Structures

This structure is designed to be used as part of a communication protocol with a server. The [MS-AADT] protocol and the [MS-ADR] protocol both use this structure for parts of their structure content.

This structure can be used in conjunction with the Conceptual Schema Definition File Format ([MC-CSDL]) or the Store Schema Definition Language File Format ([MS-SSDL]) for describing a complete table schema, as described in section 1.3.1.

1.5 Applicability Statement

This protocol is applicable for use as a standalone or a supporting representation and for publishing to a server a database application, which can include structures and actions to control workflow, the combining, filtering and ordering of data, and metadata to describe the database application as a whole.

This format provides interoperability with applications that create or read documents conforming to this structure.

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

This structure is an open schema. Authors of a database application can extend and comment this structure with their own attributes and sub-elements as long as they are in their own namespace. Note, however, that implementations that use this structure are not required to preserve unrecognized elements when loading and persisting.

2 Structures

2.1 Conceptual Overview

This section specifies the concepts used by structures that conform to this protocol's schema (section 5.1).

2.1.1 Tables

A table in a database application is a data structure that organizes data as rows (1) and columns (1). A table schema defines a table in terms of its columns (1) and other properties such as constraints and relationships.

This protocol defines elements and types that can be used to describe properties of a table schema such as check constraints, default constraints, indexes, and unique constraints.

This protocol also defines annotation elements and attributes that are used in conjunction with the Conceptual Schema Definition File Format ([MC-CSDL]) to describe other table schema properties such as columns and relationships.

2.1.1.1 Check Constraints

A check constraint specifies a validation condition that MUST be satisfied for data to be added or updated in a table. A check constraint can be defined either as a table-wide constraint or on a specific column (1) in the table. When a check constraint is added to a table, the constraint can optionally be tested on existing data. A check constraint can also optionally specify a message to be displayed to the user when the data violates the validation condition.

This protocol defines the **CheckConstraint** element (section 2.2.1.6) for specifying a check constraint.

2.1.1.2 Default Constraints

A default constraint on a column (1) in a table specifies a default value that is used when a new row (1) is inserted and no value was specified for that column (1).

This protocol defines the **DefaultConstraint** element (section 2.2.1.7) for specifying a default constraint.

2.1.1.3 Indexes

An index defined on one or more columns (1) in a table provides faster retrieval of data for searching, sorting and filtering based on the values in those columns (1). Indexes use a **sort order (3)** for each column (1) involved in the index, which specifies the order in which the column (1) data is indexed.

This protocol defines the **Index** element (section 2.2.1.8) for specifying an index on one or more columns (1) in a table in which the columns (1) can have duplicate values.

2.1.1.4 Unique Constraints

A unique constraint on a column (1) enforces uniqueness of its data by creating an index (section 2.1.1.3) on that column and disallowing duplicate values in that column. Unique constraints use a sort order (3) to determine the order in which the column data is indexed.

This protocol defines the **Unique** element (section 2.2.1.9) for specifying a unique constraint on a column (1) in a table.

2.1.1.5 Annotation Elements and Attributes

This protocol defines annotation elements and attributes that are to be used in conjunction with the Conceptual Schema Definition File Format ([MC-CSDL]) to fully describe a table schema (section 2.1.1), including table schema properties such as columns and relationships.

The following table specifies the different annotation elements for this protocol and their usage with the structures defined in the Conceptual Schema Definition File Format ([MC-CSDL]).

Element	Usage
Expression (section 2.2.1.10)	MUST be used as a child of the **Property** element ([MC-CSDL] section 2.1.3), only for **calculated columns**.
ExtendedAttributes (section 2.2.1.13)	If present, MUST be a child of the **Property** element ([MC-CSDL] section 2.1.3) to specify additional properties of the column (1), or a child of the **EntityType** element ([MC-CSDL] section 2.1.2) to specify additional properties of the table.

The following table specifies the different annotation attributes for this protocol and their usage with the structures defined in the Conceptual Schema Definition File Format ([MC-CSDL]). If present, each attribute MUST be used as an attribute of the **Property** element ([MC-CSDL] section 2.1.3) to define the corresponding column (1). In addition, the **ObjectId** attribute (section 2.2.2.16) can be used as an attribute of the **Association** element ([MC-CSDL] section 2.1.8).

Attribute	Usage
BoundColumn (section 2.2.2.5)	MUST be used only for a **lookup field**.
Caption (section 2.2.2.6)	MAY be used for any column (1).
CurrencyLocale (section 2.2.2.19)	MUST be used only for a column (1) that contains currency data.
CurrencySymbol (section 2.2.2.7)	MUST be used only for a column (1) that contains currency data.
DecimalPlaces (section 2.2.2.20)	MUST be used only for a column (1) that contains numeric data.
Description (section 2.2.2.8)	MAY be used for any column (1).
Direction (section 2.2.2.9)	MUST be used only for a lookup field.
DisplayColumn (section 2.2.2.10)	MUST be used only for a lookup field.
DisplayName (section 2.2.2.11)	MUST be used for all columns (1).
DisplayOrder (section 2.2.2.12)	MAY be used for any column (1).
Format (section 2.2.2.13)	MAY be used for any column (1).
LookupType (section 2.2.2.15)	MUST be used only for a lookup field.
ObjectId (section 2.2.2.16)	MAY be used for any column (1) or an **Association** element ([MC-CSDL] section 2.1.8).
ReferencedTable (section 2.2.2.17)	MUST be used only for a lookup field.

Attribute	Usage
StoreGeneratedPattern (section 2.2.2.18)	MUST be used only for a calculated column or for an identity column.
TextType (section 2.2.2.3)	MUST be used only for a column (1) that contains text data, including **hyperlinks**.
UnderlyingType (section 2.2.2.4)	MUST be used only for a column (1) that contains date or date with time data.
Values (section 2.2.2.19)	MUST be used only for a lookup field.

2.1.1.6 Linked Tables

Linked tables are tables in a database application whose table structure alone is stored in the back-end database while the table data is stored in an external location.

The schema of a linked table can also be described using the annotation elements and attributes of this protocol in conjunction with the Conceptual Schema Definition File Format ([MC-CSDL]) as specified in section 2.1.1.5. However, this protocol defines additional mandatory annotation attributes that are used exclusively for describing the schema of a linked table.

The following table specifies the different annotation attributes for this protocol and their usage with the structures defined in the Conceptual Schema Definition File Format ([MC-CSDL]) for describing a linked table.

Attribute	Usage
InternalName (section 2.2.2.2)	MUST be used as an attribute of the **EntityType** element ([MC-CSDL] section 2.1.2) and as an attribute of the **Property** element ([MC-CSDL] section 2.1.3) to specify the corresponding internal names of the linked table and its columns (1) respectively.
SiteUrl (section 2.2.2.1)	MUST be used as an attribute of the **EntityType** element ([MC-CSDL] section 2.1.2) to specify the source of the linked table.

In addition, the **ObjectId** attribute (section 2.2.2.16) can be used as an attribute of the **Association** element ([MC-CSDL] section 2.1.8).

2.1.2 Macros

A Macro is a set of logical constructs that control the flow of the Macro and commands that are performed when the Macro is called. Each command in the Macro is specified by an action (section 2.2.5.1) and zero or more arguments (section 2.2.5.2) that modify the meaning of the action. A Macro is called by an object in the application.

A named Macro, that is, a Macro that specifies a name, exists as a **database object** in the Application. Objects that reference a Macro can specify either the name of a named Macro or a list of inline commands. Macro commands that are specified inline in an object specify an embedded Macro.

Macros are run in one of two contexts, as specified in the next two sections.

2.1.2.1 User Interface Macros

The commands of a User Interface Macro are performed at the level of the graphical user interface of the application. If a User Interface Macro encounters any error during runtime, then the Macro will stop running.

2.1.2.2 Data Macros

The commands of a Data Macro are performed at the level of the data of the application. If a Data Macro encounters any error during runtime, including the ones raised by the **RaiseError** action (section 2.2.5.1.11), then the Macro will stop and rollback, as if no commands in the Macro had been performed.

There are two kinds of Data Macros. A Named Data Macro is a standalone database object with the name specified, and it is explicitly called through the **RunDataMacro** action (section 2.2.5.1.13). An Event Data Macro is embedded in a Table (section 2.1.1), and it is triggered once a specified data event occurs on the table.

2.1.2.3 Variables

Macros can reference the following types of variables.

2.1.2.3.1 User Interface Macro Variables

User Interface Macro variables are identified by name. They can be referenced by any User Interface Macro or any control in the database application. A User Interface Macro variable is created when the first **SetVariable** action (section 2.2.5.1.20) that references its name is performed and exists until the end of the application **session (2)**. These variables are not used in Data Macros.

2.1.2.3.2 Local Variables

Local variables are identified by name. They can be referenced only by the Macro that created them. A local variable is created when the first **SetLocalVar** action (section 2.2.5.1.17) that references its name is performed and exists until the Macro that created it has finished running. These variables are not used in User Interface Macros.

2.1.2.3.3 Return Variables

Return variables are identified by name. A return variable is created when the first **SetReturnVar** action (section 2.2.5.1.19) that references its name is performed, and it is available only in a standalone Named Data Macro (section 2.1.2.2). It exists until the standalone Named Data Macro has finished running. The value of a return variable is available to the caller of the Named Data Macro.

2.1.3 Queries

A Query in this protocol defines a way to build a query to be used as an object in a database application. It is a subset of the functionality defined by the SELECT statement in Transact-SQL ([MSDN-TSQL-Ref]).

A Query can be used to filter, order or group data. It can also be used to bring together data from more than one **input source**. Queries can take input in the form of parameters, which can be included in the output columns (1) as part of an expression or used to filter the query results. Queries can use expressions in output columns, in restrictions, in groupings, in orderings or in group

restrictions to perform calculations or formatting on data. Queries can also group data and filter these groups according to the data in those groups.

Queries include components that map to the following clauses of a SELECT statement as expressed by Transact-SQL ([MSDN-TSQL-Ref]). Restrictions on what is permitted for each Query component are described in the linked sections in this document.

Transact-SQL Clause	Query AXL
SELECT	**Results** (section 2.2.3.9) and **Parameters** (section 2.2.3.13)
DISTINCT	**Distinct** attribute of **Query** (section 2.2.3.18)
TOP	**Top Rows** (section 2.2.3.76) and **Top Percent** (section 2.2.3.77)
FROM	**References** (section 2.2.3.7) and **Joins** (section 2.2.3.11)
WHERE	**Restriction** child element of **Query** (section 2.2.3.18)
GROUP BY	**Groups** (section 2.2.3.74)
HAVING	**GroupRestriction** child element of **Query** (section 2.2.3.18)
ORDER BY	**Ordering** (section 2.2.3.3)

2.1.3.1 Aggregate Expression

Aggregate expressions perform a calculation on a set of values and return a single value. An aggregate expression uses an aggregate function, which is a **CT_FunctionCall** element (section 2.2.3.46) whose **Name** attribute has one of the following values:

Function Name
Avg
Count
Max
Min
StDev
Sum
Var

If a **CT_ExpressionContents** element (section 2.2.3.45) has at least one descendant that is an aggregate function, then the **CT_ExpressionContents** element is an aggregate expression.

2.1.4 Ignored Value

An attribute that can have any value as long as the document remains well-formed ([XML] section 2).<1>

2.2 AXL2

2.2.1 Elements

2.2.1.1 Query

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_Query** (section 2.2.3.18) element that specifies one Query (section 2.1.3) to be used as an object in a database application.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Query" type="CT_Query" wbld:cname="Query"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.2 Ordering

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_AdHocOrdering** (section 2.2.3.5) element that specifies the ordering applied to the records displayed in the form (1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Ordering" type="CT_AdHocOrdering" wbld:cname="Ordering"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.3 UserInterfaceMacros

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_UserInterfaceMacros** (section 2.2.3.41) element that specifies one or more **User Interface Macros** (section 2.1.2.1) used by a database application.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="UserInterfaceMacros" wbld:cname="UserInterfaceMacros"
type="CT_UserInterfaceMacros"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.4 DataMacros

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_DataMacros** (section 2.2.3.42) element that specifies one or more Data Macros (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="DataMacros" wbld:cname="DataMacros" type="CT_DataMacros"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.5 UserInterfaceMacro

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A CT_UserInterfaceMacro element that specifies a single **User Interface Macros** (section 2.1.2.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="UserInterfaceMacro" wbld:cname="UserInterfaceMacro"
type="CT_UserInterfaceMacro"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.6 CheckConstraint

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_CheckConstraint** (section 2.2.3.69) element that specifies a Check Constraint (section 2.1.1.1) in a Table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="CheckConstraint" wbld:cname="CheckConstraint" type="CT_CheckConstraint"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.7 DefaultConstraint

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_DefaultConstraint** (section 2.2.3.70) element that specifies a Default Constraint (section 2.1.1.2) in a Table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="DefaultConstraint" wbld:cname="DefaultConstraint"
type="CT_DefaultConstraint"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.8 Index

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_Index** (section 2.2.3.66) element that specifies an Index (section 2.1.1.3) in the Table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Index" wbld:cname="Index" type="CT_Index"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.9 Unique

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_Unique** (section 2.2.3.67) element that specifies a Unique Constraint (section 2.1.1.4) in a Table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Unique" wbld:cname="Unique" type="CT_Unique"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.10 Expression

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_ExpressionContents** (section 2.2.3.45) element that specifies an expression.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.11 EventDataMacro

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_EventDataMacro** (section 2.2.3.80) element that specifies an Event Data Macro (section 2.1.2.2) on a Table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="EventDataMacro" wbld:cname="EventDataMacro" type="CT_EventDataMacro"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.12 FormDef

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A CT_FormDef element that specifies a form (1) (section 1.3.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="FormDef" wbld:cname="FormDef" type="CT_FormDef"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.13 ExtendedAttributes

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_ExtendedAttributes** (section 2.2.3.83) element that specifies additional properties of a column (1) or table as name/value pairs.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="ExtendedAttributes" wbld:cname="ExtendedAttributes"
type="CT_ExtendedAttributes"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.14 DataMacro

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_DataMacro** (section 2.2.3.44) element that specifies a Data Macro (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="DataMacro" wbld:cname="DataMacro" type="CT_DataMacro"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.1.15 PropertyRef

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **CT_PropertyRef** element (section 2.2.3.87) that specifies a reference to a column (1) in a table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="PropertyRef" wbld:cname="PropertyRef" type="CT_PropertyRef"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2 Attributes

2.2.2.1 SiteURL

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **anyURI** ([XMLSCHEMA2] section 3.2.17) attribute that specifies the **Uniform Resource Locator (URL)** of a **site (1)** that is the source of a linked table (section 2.1.1.6).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="SiteURL" wbld:cname="SiteURL" type="xsd:anyURI"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.2 InternalName

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of a linked table (section 2.1.1.6) or the name of a **field (3)** in a linked table (section 2.1.1.6).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="InternalName" wbld:cname="InternalName" type="ST_ObjectName"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.3 TextType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_TextType** (section 2.2.4.8) attribute that specifies the layout or format of a column (1) that stores text.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="TextType" wbld:cname="TextType" type="ST_TextType"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.4 UnderlyingType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_FieldType** (section 2.2.4.6) attribute that specifies whether the underlying data type for a column (1) of type Date/Time is Date or Date/Time.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="UnderlyingType" wbld:cname="UnderlyingType" type="ST_FieldType"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.5 BoundColumn

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the column (1) in the data source (1) of a lookup field that contains the data stored in the lookup field.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="BoundColumn" wbld:cname="BoundColumn" type="ST_ObjectName"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.6 Caption

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_LongString** attribute (section 2.2.4.4) that specifies a **caption** or display name.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="Caption" type="ST_LongString" wbld:cname="Caption"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.7 CurrencySymbol

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies a currency symbol.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="CurrencySymbol" wbld:cname="CurrencySymbol" type="xsd:string"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.8 Description

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies a description of a column (1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="Description" wbld:cname="Description" type="xsd:string"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.9 Direction

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_SortDirection** attribute (section 2.2.4.7) that specifies an ordering in a sort operation.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="Direction" wbld:cname="Direction" type="ST_SortDirection"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.10 DisplayColumn

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the column (1) in the data source (1) of a lookup field that contains the data displayed by controls that are bound to the lookup field.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="DisplayColumn" wbld:cname="DisplayColumn" type="ST_ObjectName"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.11 DisplayName

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the **display name** for a column (1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="DisplayName" wbld:cname="DisplayName" type="ST_ObjectName"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.12 DisplayOrder

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **unsignedLong** ([XMLSCHEMA1] section 3.3.21) attribute that specifies the index in the display order in which this column (1) is normally displayed.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="DisplayOrder" wbld:cname="DisplayOrder" type="xsd:unsignedLong"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.13 Format

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_Format** attribute (section 2.2.4.13) that specifies a format for displaying a column (1) value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="Format" wbld:cname="Format" type="ST_Format"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.14 LookupType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_LookupType** attribute (section 2.2.4.10) that specifies the type of data source (1) for a lookup field.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="LookupType" wbld:cname="LookupType" type="ST_LookupType"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.15 ObjectId

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an identifier for an object.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="ObjectId" wbld:cname="ObjectId" type="ST_ObjectName"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.16 ReferencedTable

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the table name that contains the source column (1) for a lookup field.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="ReferencedTable" wbld:cname="ReferencedTable" type="ST_ObjectName"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.17 StoreGeneratedPattern

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **TStoreGeneratedPattern** attribute (section 2.2.4.9) that specifies the data generation pattern for a column (1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="StoreGeneratedPattern" wbld:cname="StoreGeneratedPattern"
type="TStoreGeneratedPattern"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.18 Values

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies a list of items that are the source values for a lookup field. Each value MUST be enclosed in double quotes and separated from the next value by a semicolon.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="Values" wbld:cname="Values" type="xsd:string"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.19 CurrencyLocale

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies a locale for formatting a currency value, which is defined by a language identifier and a region identifier.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="CurrencyLocale" wbld:cname="CurrencyLocale" type="xsd:string"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.2.20 DecimalPlaces

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

An **ST_DecimalPlaces** (section 2.2.4.17) attribute that specifies the number of decimal places used to display a column (1) value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this attribute.

```
<xsd:attribute name="DecimalPlaces" type="ST_DecimalPlaces" wbld:cname="DecimalPlaces"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3 Complex Types

2.2.3.1 CT_Expression

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query, CT_ForEachRecordData, CT_LookupRecordData, CT_DataIf, CT_DataElseIf, CT_UserInterfaceIf, CT_UserInterfaceElseIf

Specifies a single expression.

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies the expression.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Expression">
  <xsd:sequence>
    <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.2 CT_Order

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Ordering

Specifies a single order to apply to a **CT_Query** (section 2.2.3.18).

Attributes:

Direction: An **ST_SortDirection** (section 2.2.4.7) attribute that specifies the direction of the order.

Name: An **ST_LongString** (section 2.2.4.4) attribute that specifies one column (1) from the input source specified by the **Source** attribute. The column (1) MUST be included in the **CT_Results** element (section 2.2.3.9) of the parent **CT_Query** (section 2.2.3.18). If the **CT_Result** (section 2.2.3.8) element specifying the column (1) has an **Alias** attribute, then this attribute MUST have that value. The data from that column (1) is ordered in the query **CT_Results** according to **Direction**.

Source: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name or alias of one of the input sources to the query. The input source MUST be listed in the **CT_References** (section 2.2.3.7) collection of the parent **CT_Query** (section 2.2.3.18). If the **CT_Reference** (section 2.2.3.6) element specifying the input source has an **Alias** attribute, then this attribute MUST have that value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Order">
  <xsd:attribute name="Name" type="ST_LongString" use="required" wbld:cname="Name"/>
  <xsd:attribute name="Source" type="ST_ObjectName" use="required" wbld:cname="Source"/>
  <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.3 CT_Ordering

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Container type for all of the ordering specified in a **CT_Query** (section 2.2.3.18). MUST contain at least one and no more than 255 **Order** or **OrderExpression** elements. The final query results are ordered such that the first listed **Order** or **OrderExpression** takes precedence across the entire result set, with the second listed **Order** or **OrderExpression** affecting the arrangement of rows (1) within the limits imposed by the first ordering, and so on.

Child Elements:

Order: A **CT_Order** (section 2.2.3.2) element that specifies one column (1) from the **CT_Results** (section 2.2.3.9) of the query and a direction. Query results are sorted on that column (1) in the given direction.

OrderExpression: A **CT_OrderExpression** (section 2.2.3.71) element that specifies an expression and a direction. Query results are sorted by the value of that expression for each row (1) in the given direction.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Ordering">
  <xsd:choice maxOccurs="255">
    <xsd:element name="Order" wbld:cname="Order" type="CT_Order"/>
```

```
    <xsd:element name="OrderExpression" wbld:cname="OrderExpression"
type="CT_OrderExpression"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.4 CT_AdHocOrder

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_AdHocOrdering

Specifies one column (1) to order.

Attributes:

Direction: An **ST_SortDirection** (section 2.2.4.7) attribute that specifies the sort order (3) as either ascending or descending.

Name: An **ST_LongString** (section 2.2.4.4) attribute that specifies the column (1) to which the order is applied.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_AdHocOrder">
  <xsd:attribute name="Name" type="ST_LongString" use="required" wbld:cname="Name"/>
  <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.5 CT_AdHocOrdering

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: Ordering

The **CT_AdHocOrdering** complex type is a container type for the **CT_AdHocOrder** (section 2.2.3.4) and **CT_OrderExpression** (section 2.2.3.71) elements. Elements of this type MUST contain at least one and no more than 255 **Order** or **OrderExpression** elements.

Child Elements:

Order: A **CT_AdHocOrder** element that specifies one column (1), and a direction to order.

OrderExpression: A **CT_OrderExpression** element that specifies an expression, and a direction to order.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_AdHocOrdering">
```

```
  <xsd:choice maxOccurs="255">
    <xsd:element name="Order" wbld:cname="Order" type="CT_AdHocOrder"/>
    <xsd:element name="OrderExpression" wbld:cname="OrderExpression"
type="CT_OrderExpression"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.6 CT_Reference

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_References

Specifies one input source to the parent **CT_Query** (section 2.2.3.18).

Child Elements:

ReferenceParameters: A **CT_Parameters** element (section 2.2.3.13) that specifies parameters of the input source. If a **CT_Parameter** (section 2.2.3.12) specified in this element has the same **Name** attribute as a **CT_Parameter** in the **Parameters** element of the parent **CT_Query** (section 2.2.3.18) or in the **ReferenceParameters** element of a different **CT_Reference** in the same parent **CT_Query**, their **Type** attributes MUST be the same.

If the parent **CT_Query** is used as an input source to another query, the parameters in **CT_Query** and the parameters in **ReferenceParameters** in order of input sources MUST be listed in **ReferenceParameters** of the **References** element in another query. While adding the **ReferenceParameters** of input sources to the list, the parameter MUST be skipped if a **CT_Parameter** with the same **Name** has already been added.

Attributes:

Alias: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an alias for the input source. This alias MUST be used throughout the Query object to refer to the input source as if it had been named this alias. The value MUST be unique among all **Alias** attribute values under the **CT_References** collection (section 2.2.3.7).

Source: An **ST_ObjectName** attribute that specifies a table or a named query (section 2.1.3) that contains data included in the query **CT_Results** (section 2.2.3.9).

Type: An **ST_QuerySourceType** attribute (section 2.2.4.18) that specifies the type of the input source.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Reference">
  <xsd:all>
    <xsd:element name="ReferenceParameters" minOccurs="0" type="CT_Parameters"
wbld:cname="ReferenceParameters"/>
  </xsd:all>
  <xsd:attribute name="Source" type="ST_ObjectName" use="required" wbld:cname="Source"/>
  <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
  <xsd:attribute name="Type" type="ST_QuerySourceType" default="Table" wbld:cname="Type"
use="optional"/>
```

```
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.7 CT_References

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Container type for the input sources to a **CT_Query** (section 2.2.3.18). MUST contain one or more **Reference** elements.

Child Elements:

Reference: A **CT_Reference** (section 2.2.3.6) element that specifies a Table (section 2.1.1) or Query (section 2.1.3) that is an input source to the **CT_Query** (section 2.2.3.18).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_References">
  <xsd:sequence>
    <xsd:element name="Reference" maxOccurs="unbounded" wbld:cname="Reference"
type="CT_Reference"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.8 CT_Result

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Results

Specifies one or more columns (1) of data to be included in the **CT_Results** (section 2.2.3.9) of a **CT_Query** (section 2.2.3.18).

The specification is for one column (1) of data from a single input source (that is, the **All** attribute is false and the **Name** and **Source** attributes are present), all columns (1) from a single input source (that is, **All** is true and **Source** is present), all columns (1) from all input sources (that is, **All** is true and **Source** is not present), or a single expression (that is, an **Expression** child element is present).

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies a single expression to be included in the query results. The expression MUST be evaluated per row (1) in the results. This element MUST NOT be present if the **Source** or **Name** attribute is present or if the **All** attribute is true. When this attribute is present, the **Alias** attribute MUST also be present.

Attributes:

Alias: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies an **alias** for a column (1) in the query results. This attribute MUST NOT be present if the **All** attribute is set to true. This attribute MUST be present if an **Expression** child element is present. The value MUST be unique among all **Alias** attribute values under the **CT_Results** (section 2.2.3.9) collection.

If the same single column (1) of data is specified by more than one **Result** element in a **CT_Results** (section 2.2.3.9) collection, not including **Result** elements where the **All** attribute is true, then either all or all but one **Result** element that specifies the column by including both a **Source** attribute and a **Name** attribute MUST have an **Alias** attribute present.

All: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute. If this attribute is true and the **Source** attribute is present, then the query results contain all columns (1) from the input source indicated by the **Source** attribute. If this attribute is true and the **Source** attribute is not present, then the query results contain all columns (1) from all input sources listed in the **CT_References** (section 2.2.3.7) collection. When this attribute is true, the **Expression** child element, the **Name** attribute, and the **Alias** attribute MUST NOT be present.

Name: An **ST_LongString** (section 2.2.4.4) attribute that specifies the name of one column (1) from the input source specified by the **Source** attribute. The data from this column (1) is included in the query results. This attribute MUST NOT be present if the **All** attribute is set to true or if an **Expression** child element is present.

Source: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name or alias of one of the input sources to the query. This attribute MUST NOT be present if an **Expression** child element is present. The input source MUST be listed in the **CT_References** (section 2.2.3.7) collection of the query. If the **CT_Reference** (section 2.2.3.6) element specifying the input source contains an **Alias** value, then this attribute MUST be that value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Result">
  <xsd:all>
    <xsd:element name="Expression" type="CT_ExpressionContents" minOccurs="0"
wbld:cname="Expression"/>
  </xsd:all>
  <xsd:attribute name="Source" type="ST_ObjectName" wbld:cname="Source" use="optional"/>
  <xsd:attribute name="Name" type="ST_LongString" wbld:cname="Name" use="optional"/>
  <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
  <xsd:attribute name="All" type="xsd:boolean" default="false" wbld:cname="All"
use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.9 CT_Results

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Container element for the results in a **CT_Query** (section 2.2.3.18). MUST include at least one and no more than 255 **Property** elements.

Child Elements:

Property: A **CT_Result** (section 2.2.3.8) element that specifies one or more columns (1) of data to include in the **CT_Query** (section 2.2.3.18) results.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Results">
  <xsd:sequence>
    <xsd:element name="Property" maxOccurs="255" wbld:cname="Property" type="CT_Result"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.10 CT_Join

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Joins

Description of a single join in a **CT_Query** (section 2.2.3.18). A join MUST NOT have the same pair of **Left** and **Right** attribute values as another join in the same **CT_Query** (section 2.2.3.18).

Attributes:

Left: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the left input source. The input source MUST be listed in the **CT_References** (section 2.2.3.7) collection of the parent **CT_Query** element (section 2.2.3.18). If the **CT_Reference** element (section 2.2.3.6) that specifies the left input source specifies an **Alias** attribute, then this attribute MUST be that value.

LeftProperty: An **ST_LongString** (section 2.2.4.4) attribute that specifies the column (1) in the left input source on which to join.

Right: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the right input source. The input source MUST be listed in the **CT_References** (section 2.2.3.7) collection of the parent **CT_Query** element (section 2.2.3.18). If the **CT_Reference** element (section 2.2.3.6) that specifies the right input source includes an **Alias** attribute, then this attribute MUST be that value.

RightProperty: An **ST_LongString** (section 2.2.4.4) attribute that specifies the column (1) in the right input source on which to join.

Type: An **ST_JoinDirection** (section 2.2.4.5) attribute that specifies the direction of the join.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Join">
  <xsd:attribute name="Left" type="ST_ObjectName" use="required" wbld:cname="Left"/>
  <xsd:attribute name="Right" type="ST_ObjectName" use="required" wbld:cname="Right"/>
  <xsd:attribute name="LeftProperty" type="ST_LongString" use="required"
wbld:cname="LeftProperty"/>
  <xsd:attribute name="RightProperty" type="ST_LongString" use="required"
wbld:cname="RightProperty"/>
  <xsd:attribute name="Type" type="ST_JoinDirection" default="Inner" wbld:cname="Type"
use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.11 CT_Joins

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Container type for the joins in a **CT_Query** (section 2.2.3.18). MUST contain one or more **Join** elements.

Child Elements:

Join: A **CT_Join** (section 2.2.3.10) element that specifies a single join in the **CT_Query** (section 2.2.3.18).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Joins">
  <xsd:sequence>
    <xsd:element name="Join" wbld:cname="Join" maxOccurs="unbounded" type="CT_Join"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.12 CT_Parameter

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Parameters

Specifies a single parameter of a **CT_Reference** (section 2.2.3.6) or a **CT_Query** (section 2.2.3.18).

Attributes:

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the parameter.

Type: An **ST_FieldType** (section 2.2.4.6) attribute that specifies the data type of the parameter.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Parameter">
  <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
  <xsd:attribute name="Type" type="ST_FieldType" use="required" wbld:cname="Type"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.13 CT_Parameters

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Reference, CT_Query

Container type for the parameters in a **CT_Reference** (section 2.2.3.6) or in a **CT_Query** (section 2.2.3.18). MUST contain one or more **Parameter** elements.

Child Elements:

Parameter: A **CT_Parameter** (section 2.2.3.12) element that specifies one parameter of a **CT_Reference** (section 2.2.3.6) or of a **CT_Query** (section 2.2.3.18).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Parameters">
  <xsd:sequence>
    <xsd:element name="Parameter" maxOccurs="unbounded" wbld:cname="Parameter"
type="CT_Parameter"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.14 CT_ParameterDefinition

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_ParameterDefinitions

Specifies one parameter that is accepted by a Data Macro (section 2.1.2.2)

Attributes:

Description: An **Ignored Value** (section 2.1.4).

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the parameter. The value of this attribute MUST be unique among all other elements of this type **CT_ParameterDefinition** in the same Data Macro (section 2.1.2.2).

Type: An **ST_FieldType** (section 2.2.4.6) attribute that specifies the data type of the parameter.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ParameterDefinition">
  <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
  <xsd:attribute name="Description" type="ST_LongString" wbld:cname="Description"
use="optional"/>
  <xsd:attribute name="Type" type="ST_FieldType" use="required" wbld:cname="Type"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.15 CT_ParameterDefinitions

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacro

Container type for the parameter definitions in a Data Macro (section 2.1.2.2). MUST contain one or more **Parameter** elements.

Child Elements:

Parameter: A **CT_ParameterDefinition** (section 2.2.3.14) element that specifies a parameter supported by the Data Macro (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ParameterDefinitions">
  <xsd:sequence>
    <xsd:element name="Parameter" maxOccurs="unbounded" wbld:cname="Parameter"
type="CT_ParameterDefinition"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.16 CT_ParameterValue

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_ParameterValues

Specifies one parameter to be passed into a macro action (section 2.2.5.1).

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies the value of the parameter.

Attributes:

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the parameter.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ParameterValue">
  <xsd:sequence>
    <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
  </xsd:sequence>
  <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.17 CT_ParameterValues

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Action, CT_ForEachRecordData, CT_LookupRecordData, CT_CreateRecordData

Container type for the parameter values to be passed into a macro action (section 2.2.5.1). MUST contain one or more **Parameter** elements or **OutputParameter** elements.

Child Elements:

OutputParameter: A **CT_OutputParameter** (section 2.2.3.75) element that specifies the name of a local variable to store the output parameter value. This element MUST NOT appear in any **CT_ParameterValues** other than inside **CT_CreateRecord** (section 2.2.3.26) or **RunDataMacro** action (section 2.2.5.1.13). This element MUST NOT appear more than once inside **CT_CreateRecord** (section 2.2.3.26).

Parameter: A **CT_ParameterValue** (section 2.2.3.16) element that specifies a single value to be passed into a macro action (section 2.2.5.1). The value of the **Name** attribute of each **Parameter** element MUST be unique within the same **CT_ParameterValues**. This element MUST NOT appear inside **CT_ParameterValues** referred by **CT_CreateRecord** (section 2.2.3.26).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ParameterValues">
  <xsd:sequence>
    <xsd:element name="Parameter" minOccurs="0" maxOccurs="unbounded" wbld:cname="Parameter"
type="CT_ParameterValue"/>
    <xsd:element name="OutputParameter" minOccurs="0" maxOccurs="unbounded"
wbld:cname="OutputParameter" type="CT_OutputParameter"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.18 CT_Query

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_FormDef, Query

Specifies a Query (section 2.1.3).

Child Elements:

GroupRestriction: A **CT_Expression** (section 2.2.3.1) element that specifies a filter for groups in the query. All groups included in the query results MUST meet the criteria defined by the group restriction expression.

Groups: A **CT_Groups** (section 2.2.3.74) element that specifies the groups into which records will be placed in the query results.

Joins: A **CT_Joins** (section 2.2.3.11) element that specifies all joins in the query.

Ordering: A **CT_Ordering** (section 2.2.3.3) element that specifies the order of the query results.

Parameters: A **CT_Parameters** (section 2.2.3.13) element that specifies all parameters in the query.

References: A **CT_References** (section 2.2.3.7) element that specifies all input sources for the query.

Restriction: A **CT_Expression** (section 2.2.3.1) element that specifies a filter for the query. All rows (1) included in the query results MUST meet the criteria defined by the restriction expression.

Results: A **CT_Results** (section 2.2.3.9) element that specifies the columns (1) included in the result of running the query.

TopPercent: A **CT_TopPercent** (section 2.2.3.77) element that specifies the limit on the percentage of records to be returned as query results.

TopRows: A **CT_TopRows** (section 2.2.3.76) element that specifies the limit on the number of records to be returned as query results.

Attributes:

Distinct: A boolean ([XMLSCHEMA2] section 3.2.2) attribute. If this attribute is true, then the query results MUST contain only unique rows (1).

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the query. SHOULD be present when **CT_Query** (section 2.2.3.18) is used as a child element for **CT_FormDef** (section 2.2.3.81). SHOULD NOT be present when the CT_Query (section 2.2.3.18) is the top element of the document.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Query">
  <xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element name="TopRows" type="CT_TopRows" wbld:cname="TopRows"/>
      <xsd:element name="TopPercent" type="CT_TopPercent" wbld:cname="TopPercent"/>
    </xsd:choice>
    <xsd:element name="Parameters" minOccurs="0" type="CT_Parameters"
wbld:cname="Parameters"/>
    <xsd:element name="References" type="CT_References" wbld:cname="References"/>
    <xsd:element name="Results" type="CT_Results" wbld:cname="Results"/>
    <xsd:element name="Joins" minOccurs="0" type="CT_Joins" wbld:cname="Joins"/>
    <xsd:element name="Restriction" type="CT_Expression" minOccurs="0"
wbld:cname="Restriction"/>
    <xsd:element name="Groups" minOccurs="0" type="CT_Groups" wbld:cname="Groups"/>
    <xsd:element name="GroupRestriction" minOccurs="0" type="CT_Expression"
wbld:cname="GroupRestriction"/>
    <xsd:element name="Ordering" minOccurs="0" type="CT_Ordering" wbld:cname="Ordering"/>
  </xsd:sequence>
  <xsd:attribute name="Name" type="ST_ObjectName" wbld:cname="Name" use="optional"/>
  <xsd:attribute name="Distinct" type="xsd:boolean" wbld:cname="Distinct" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.19 CT_Argument

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Action

Specifies an argument of a macro action (section 2.2.5.1).

Attributes:

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the argument.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Argument">
  <xsd:simpleContent>
    <xsd:extension base="ST_LongString">
      <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.20 CT_Action

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements, CT_UserInterfaceMacroStatements

Specifies a macro action (section 2.2.5.1).

Child Elements:

Argument: A **CT_Argument** (section 2.2.3.19) element that specifies an argument for the action where the value of the argument is a string.

ExpressionArgument: A **CT_ExpressionArgument** (section 2.2.3.78) element that specifies an argument for the action where the value of the argument is an expression.

Parameters: A **CT_ParameterValues** (section 2.2.3.17) element that specifies the parameters to the action. MUST NOT be present except as part of the **RunDataMacro** action (section 2.2.5.1.13), the **ChangeView** action (section 2.2.5.1.1), or the **OpenPopup** action (section 2.2.5.1.10).

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the action. MUST be one of the actions specified by section 2.2.5.1.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Action">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Argument" wbld:cname="Argument" type="CT_Argument"/>
      <xsd:element name="ExpressionArgument" wbld:cname="ExpressionArgument"
type="CT_ExpressionArgument"/>
    </xsd:choice>
    <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
  </xsd:sequence>
```

```
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
  </xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.21 CT_ForEachRecordData

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_ForEachRecord

Specifies a set of records.

Child Elements:

Parameters: A **CT_ParameterValues** (section 2.2.3.17) element that specifies the parameters to the query (section 2.1.3) specified by the **Reference** element. MUST NOT be present if the **Reference** element specifies a Table (section 2.1.1).

Reference: An **ST_ObjectName** (section 2.2.4.1) element that specifies a Table (section 2.1.1) or a Query (section 2.1.3).

WhereCondition: A **CT_Expression** (section 2.2.3.1) element that specifies an expression that selects a set of records from the records in the object specified by the **Reference** element.

Attributes:

Alias: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies a name for this set of records.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ForEachRecordData">
  <xsd:sequence>
    <xsd:element name="Reference" type="ST_ObjectName" wbld:cname="Reference"/>
    <xsd:element name="WhereCondition" minOccurs="0" type="CT_Expression"
wbld:cname="WhereCondition"/>
    <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
  </xsd:sequence>
  <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.22 CT_ForEachRecord

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements

Specifies a set of commands that are performed for each record in a specified set of records.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) element that is a child of a **CT_CreateRecord** (section 2.2.3.26) or **CT_EditRecord** (section 2.2.3.28) element.

Child Elements:

Data: A **CT_ForEachRecordData** (section 2.2.3.21) element that specifies the set of records on which to perform the commands.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies a set of commands to perform on each record.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ForEachRecord">
  <xsd:sequence>
    <xsd:element name="Data" wbld:cname="Data" type="CT_ForEachRecordData"/>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.23 CT_LookupRecordData

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_LookupRecord

Specifies a single record.

Child Elements:

Parameters: A **CT_ParameterValues** (section 2.2.3.17) element that specifies the parameters to the query (section 2.1.3) specified by the **Reference** element. MUST NOT be present if the **Reference** element specifies a Table (section 2.1.1).

Reference: An ST_ObjectName element that specifies a Table (section 2.1.1) or a Query (section 2.1.3).

WhereCondition: A **CT_Expression** (section 2.2.3.1) element that specifies an expression that selects a single record from the object specified by the **Reference** element

Attributes:

Alias: An **ST_ObjectName** section (2.2.4.1) attribute that specifies a name for this record.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_LookupRecordData">
  <xsd:sequence>
    <xsd:element name="Reference" type="ST_ObjectName" wbld:cname="Reference"/>
    <xsd:element name="WhereCondition" minOccurs="0" type="CT_Expression"
wbld:cname="WhereCondition"/>
```

```
    <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
  </xsd:sequence>
  <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.24 CT_LookupRecord

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements

Specifies a set of commands that are performed for a single record.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) element that is a child of a **CT_CreateRecord** (section 2.2.3.26) or **CT_EditRecord** (section 2.2.3.28) element.

Child Elements:

Data: A **CT_LookupRecordData** (section 2.2.3.23) element that specifies a single record.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies a set of commands to perform on a single record.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_LookupRecord">
  <xsd:sequence>
    <xsd:element name="Data" wbld:cname="Data" type="CT_LookupRecordData"/>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.25 CT_CreateRecordData

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_CreateRecord

Specifies a Table (section 2.1.1) in which to create a new record.

Child Elements:

Parameters: A **CT_ParameterValues** (section 2.2.3.17) element that specifies the local variable to store the identifier of the newly created record.

Reference: An **ST_ObjectName** (section 2.2.4.1) element that specifies a Table (section 2.1.1).

Attributes:

Alias: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies a name for the newly created record in the Table (section 2.1.1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_CreateRecordData">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="Reference" type="ST_ObjectName" wbld:cname="Reference"/>
    </xsd:choice>
    <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
  </xsd:sequence>
  <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.26 CT_CreateRecord

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements

Specifies a set of commands that are performed on a newly created record.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) element that is a child of a **CT_EditRecord** (section 2.2.3.28) element or another **CT_CreateRecord** (section 2.2.3.26) element.

Child Elements:

Data: A **CT_CreateRecordData** (section 2.2.3.25) element that specifies the data source (1) in which a record will be created.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies a set of commands to perform on the new record.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_CreateRecord">
  <xsd:sequence>
    <xsd:element name="Data" wbld:cname="Data" type="CT_CreateRecordData"/>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
```

```
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.27 CT_EditRecordData

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_EditRecord

Specifies a single record.

Attributes:

Alias: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name given to a single record. This attribute MUST be the same as the **Alias** attribute of a **CT_ForEachRecordData** (section 2.2.3.21) element, the **Alias** attribute of a **CT_LookupRecordData** (section 2.2.3.23) element, or the name of the parent Table (section 2.1.1) of the Data Macro (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_EditRecordData">
  <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.28 CT_EditRecord

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements

Specifies a set of commands to run on a single record.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) that is a child of a **CT_CreateRecord** (section 2.2.3.26) element or of another **CT_EditRecord** (section 2.2.3.28) element.

Child Elements:

Data: A **CT_EditRecordData** (section 2.2.3.27) element that specifies a single record.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies a set of commands to perform on a single record.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_EditRecord">
```

```
    <xsd:sequence>
      <xsd:element name="Data" type="CT_EditRecordData" wbld:cname="Data"/>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
  </xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.29 CT_UserInterfaceIf

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceConditionalBlock

Specifies a branch of a conditional block.

Child Elements:

Condition: A **CT_Expression** (section 2.2.3.1) element that specifies a condition, which evaluates to true or false.

Statements: A **CT_UserInterfaceMacroStatements** (section 2.2.3.37) element that specifies the set of commands that are performed if the expression in the **Condition** element evaluates to true.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceIf">
  <xsd:sequence>
    <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
    <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.30 CT_UserInterfaceElseIf

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceConditionalBlock

Specifies a branch of a conditional block. This element MUST follow an **If** or another **ElseIf** element in the parent element.

Child Elements:

Condition: A **CT_Expression** (section 2.2.3.1) element that specifies a condition, which evaluates to true or false.

Statements: A **CT_UserInterfaceMacroStatements** (section 2.2.3.37) element that specifies the set of commands that are performed if the expression in the **Condition** element evaluates to true and the **Condition** elements in the preceding **If** and **ElseIf** elements in the parent element evaluate to false.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceElseIf">
  <xsd:sequence>
    <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
    <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.31 CT_UserInterfaceElse

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceConditionalBlock

Specifies a conditional branch. This element MUST follow an **If** or **ElseIf** element in the parent element.

Child Elements:

Statements: A **CT_UserInterfaceMacroStatements** (section 2.2.3.37) element that specifies the set of commands that are performed if the **Condition** elements of the **If** and **ElseIf** elements in the parent element evaluate to false.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceElse">
  <xsd:sequence>
    <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.32 CT_DataIf

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataConditionalBlock

Specifies a branch of a conditional block.

Child Elements:

Condition: A **CT_Expression** (section 2.2.3.1) element that specifies a condition, which evaluates to true or false.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies the set of commands that are performed if the expression in the **Condition** element evaluates to true.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataIf">
  <xsd:sequence>
    <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.33 CT_DataElseIf

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataConditionalBlock

Specifies a branch of a conditional block. This element MUST follow an **If** or another **ElseIf** element in the parent element.

Child Elements:

Condition: A **CT_Expression** (section 2.2.3.1) element that specifies a condition, which evaluates to true or false.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies the set of commands that are performed if the expression in the **Condition** element evaluates to true and the **Condition** elements in the preceding **If** and **ElseIf** elements in the parent element evaluate to false.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataElseIf">
  <xsd:sequence>
    <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.34 CT_DataElse

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataConditionalBlock

Specifies a conditional branch. This element MUST follow an **If** or **ElseIf** element in the parent element.

Child Elements:

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies the set of commands that are performed if the **Condition** elements of the preceding **If** and **ElseIf** elements in the parent element evaluate to false.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataElse">
  <xsd:sequence>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.35 CT_UserInterfaceConditionalBlock

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceMacroStatements

Specifies a conditional branch in a User Interface Macro (section 2.1.2.1). At runtime, if the expression in the **Condition** element of the **If** element evaluates to true, the macro continues performing the commands within the **Statements** child element of the **If** element and all other branches are skipped. Otherwise the expression in the **Condition** element of each **ElseIf** child, if any are present, is evaluated in order until one is found that evaluates to true. If such a child is

found, the macro continues running in the **Statements** child element of that **ElseIf** element and all other branches are skipped. If the conditions of the **If** child element and all **ElseIf** child elements evaluate to false and an **Else** element is present, the macro continues running in the **Statements** child of the **Else** element.

Child Elements:

Else: A **CT_UserInterfaceElse** (section 2.2.3.31) element that specifies a branch to be taken if none of the conditions of the other child elements evaluate to true.

ElseIf: A **CT_UserInterfaceElseIf** (section 2.2.3.30) element that specifies a subsequent condition and logical branch to be taken if the **If** branch and previous **ElseIf** branches were not taken.

If: A **CT_UserInterfaceIf** (section 2.2.3.29) element that specifies the first possible condition and logical branch. MUST be present.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceConditionalBlock">
  <xsd:sequence>
    <xsd:element name="If" wbld:cname="If" type="CT_UserInterfaceIf"/>
    <xsd:element name="ElseIf" minOccurs="0" maxOccurs="unbounded" wbld:cname="ElseIf"
type="CT_UserInterfaceElseIf"/>
    <xsd:element name="Else" minOccurs="0" wbld:cname="Else" type="CT_UserInterfaceElse"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.36 CT_DataConditionalBlock

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements

Specifies a conditional branch in a Data Macro (section 2.1.2.2). At runtime, if the expression in the **Condition** element of the **If** element evaluates to true, the macro continues running in the commands within the **Statements** child element of the **If** element and all other branches are skipped. Otherwise the expression in the **Condition** element of each **ElseIf** child, if any are present, is evaluated in order until one is found that evaluates to true. If such a child is found, the macro continues running in the **Statements** child element of that **ElseIf** element and all other branches are skipped. If the conditions of the **If** child element and all **ElseIf** child elements evaluate to false and an **Else** element is present, the macro continues running in the **Statements** child of the **Else** element.

Child Elements:

Else: A **CT_DataElse** (section 2.2.3.34) element that specifies a branch to be taken if none of the conditions of the other child elements evaluate to true.

ElseIf: A **CT_DataElseIf** (section 2.2.3.33) element that specifies a subsequent condition and logical branch to be taken if the **If** branch and previous **ElseIf** branches were not taken.

If: A **CT_DataIf** (section 2.2.3.32) element that specifies the first possible condition and logical branch. MUST be present.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataConditionalBlock">
  <xsd:sequence>
    <xsd:element name="If" wbld:cname="If" type="CT_DataIf"/>
    <xsd:element name="ElseIf" minOccurs="0" maxOccurs="unbounded" wbld:cname="ElseIf"
type="CT_DataElseIf"/>
    <xsd:element name="Else" minOccurs="0" wbld:cname="Else" type="CT_DataElse"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.37 CT_UserInterfaceMacroStatements

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceIf, CT_UserInterfaceElseIf, CT_UserInterfaceMacroStatementGroup, CT_UserInterfaceElse, CT_UserInterfaceMacro

Container type that contains zero or more elements that specify the commands to be performed when a User Interface Macro (section 2.1.2.1) is called.

Child Elements:

Action: A **CT_Action** (section 2.2.3.20) element that specifies a single macro action.

Comment: An **ST_LongString** (section 2.2.4.4) element that is ignored when the commands are performed.

ConditionalBlock: A CT_UserInterfaceConditionalBlock element that specifies a conditional branch.

StatementGroup: A CT_UserInterfaceMacroStatementGroup element that specifies a user-defined grouping of macro commands.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceMacroStatements">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Action" type="CT_Action" wbld:cname="Action"/>
    <xsd:element name="Comment" type="ST_LongString" wbld:cname="Comment"/>
    <xsd:element name="ConditionalBlock" type="CT_UserInterfaceConditionalBlock"
wbld:cname="ConditionalBlock"/>
    <xsd:element name="StatementGroup" type="CT_UserInterfaceMacroStatementGroup"
wbld:cname="StatementGroup"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.38 CT_DataMacroStatements

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_ForEachRecord, CT_LookupRecord, CT_CreateRecord, CT_EditRecord, CT_DataIf, CT_DataMacroStatementGroup, CT_DataElseIf, CT_DataElse, CT_DataMacro

Container type that contains zero or more elements that specify the commands to be performed when a Data Macro (section 2.1.2.2) is called.

Child Elements:

Action: A **CT_Action** (section 2.2.3.20) element that specifies a single macro action.

Comment: An **ST_LongString** (section 2.2.4.4) element that is ignored when the commands are performed.

ConditionalBlock: A **CT_DataConditionalBlock** (section 2.2.3.36) element that specifies a conditional branch.

CreateRecord: A **CT_CreateRecord** (section 2.2.3.26) element that specifies a macro block that creates a record.

EditRecord: A **CT_EditRecord** (section 2.2.3.28) element that specifies a macro block that changes a record.

ForEachRecord: A **CT_ForEachRecord** (section 2.2.3.22) element that specifies a macro block that performs actions on multiple records.

LookUpRecord: A **CT_LookupRecord** (section 2.2.3.24) element that specifies a macro block that performs actions on a single record.

StatementGroup: A **CT_DataMacroStatementGroup** (section 2.2.3.40) element that specifies a user-defined grouping of macro commands.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataMacroStatements">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Action" type="CT_Action" wbld:cname="Action"/>
    <xsd:element name="Comment" type="ST_LongString" wbld:cname="Comment"/>
    <xsd:element name="ForEachRecord" type="CT_ForEachRecord" wbld:cname="ForEachRecord"/>
    <xsd:element name="LookUpRecord" type="CT_LookupRecord" wbld:cname="LookupRecord"/>
    <xsd:element name="CreateRecord" type="CT_CreateRecord" wbld:cname="CreateRecord"/>
    <xsd:element name="EditRecord" type="CT_EditRecord" wbld:cname="EditRecord"/>
    <xsd:element name="ConditionalBlock" type="CT_DataConditionalBlock"
wbld:cname="ConditionalBlock"/>
    <xsd:element name="StatementGroup" type="CT_DataMacroStatementGroup"
wbld:cname="StatementGroup"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.39 CT_UserInterfaceMacroStatementGroup

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceMacroStatements

Container type for macro statements with which a user can organize the actions in a macro. Has no effect on the macro commands that are performed.

Child Elements:

Statements: A **CT_UserInterfaceMacroStatements** (section 2.2.3.37) element that specifies the logic and actions to be performed in this part of the macro.

Attributes:

Collapsed: An Ignored Value (section 2.1.4)

Description: An **ST_LongString** (section 2.2.4.4) attribute that specifies a description of the enclosed block.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceMacroStatementGroup">
  <xsd:sequence>
    <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Description" type="ST_LongString" wbld:cname="Description"
use="optional"/>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.40 CT_DataMacroStatementGroup

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacroStatements

Container type for macro statements with which a user can organize the actions in a macro. Has no effect on the macro commands that are performed.

Child Elements:

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies the logic and actions to be performed in this part of the macro.

Attributes:

Collapsed: An Ignored Value (section 2.1.4).

Description: An **ST_LongString** (section 2.2.4.4) attribute that specifies a description of the enclosed block.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataMacroStatementGroup">
  <xsd:sequence>
    <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Description" type="ST_LongString" wbld:cname="Description"
use="optional"/>
  <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.41 CT_UserInterfaceMacros

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_FormDef, UserInterfaceMacros

Container type for User Interface Macros (section 2.1.2.1) on a form (1). MUST contain one or more **UserInterfaceMacro** elements.

Child Elements:

UserInterfaceMacro: A **CT_UserInterfaceMacro** (section 2.2.3.43) element that specifies a single User Interface Macro.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceMacros">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="UserInterfaceMacro" minOccurs="0" maxOccurs="unbounded"
type="CT_UserInterfaceMacro" wbld:cname="UserInterfaceMacro"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.42 CT_DataMacros

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: DataMacros

Container type for Data Macros (section 2.1.2.2). MUST contain one or more **DataMacro** elements.

Child Elements:

DataMacro: A **CT_DataMacro** (section 2.2.3.44) element that specifies a single Data Macro (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataMacros">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="DataMacro" minOccurs="0" maxOccurs="unbounded" type="CT_DataMacro"
wbld:cname="DataMacro"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.43 CT_UserInterfaceMacro

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceMacros, UserInterfaceMacro

Specifies a User Interface Macro (section 2.1.2.1).

Child Elements:

Statements: A **CT_UserInterfaceMacroStatements** (section 2.2.3.37) element that specifies the commands to be performed when the macro is called.

Attributes:

Event: An **ST_UserInterfaceObjectEvent** (section 2.2.4.12) attribute that specifies the event that will cause the macro to be called. MUST be present if the macro is embedded in a form (1); MUST NOT be present otherwise.

For: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of a control. MUST NOT appear except in a macro that is embedded in a form (1). If this attribute is not present, the macro will be called by the form itself. If present, the value MUST be the name of a control on the containing form whose type supports the event specified by the **Event** attribute.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_UserInterfaceMacro">
  <xsd:sequence>
    <xsd:element name="Statements" minOccurs="0" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="For" type="ST_ObjectName" wbld:cname="For" use="optional"/>
  <xsd:attribute name="Event" type="ST_UserInterfaceObjectEvent" wbld:cname="Event"
use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.44 CT_DataMacro

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacros, CT_EventDataMacro, DataMacro

Specifies a Data Macro (section 2.1.2.2).

Child Elements:

Parameters: A **CT_ParameterDefinitions** (section 2.2.3.15) element that specifies the list of parameters that the macro accepts when called. MUST NOT be present if the **Event** attribute is present.

Statements: A **CT_DataMacroStatements** (section 2.2.3.38) element that specifies the commands to be performed when the macro is called.

Attributes:

Event: An **ST_DataObjectEvent** (section 2.2.4.2) attribute that specifies the type of event that will cause the macro to be called.

Version: A string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the version of the Data Macro (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DataMacro">
  <xsd:sequence>
    <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterDefinitions"
wbld:cname="Parameters"/>
    <xsd:element name="Statements" minOccurs="0" type="CT_DataMacroStatements"
wbld:cname="Statements"/>
  </xsd:sequence>
  <xsd:attribute name="Event" type="ST_DataObjectEvent" wbld:cname="Event" use="optional"/>
  <xsd:attribute name="Version" type="xsd:string" wbld:cname="Version" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.45 CT_ExpressionContents

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_NamedExpression, CT_Expression, CT_ConstraintWithExpression, CT_CheckConstraint, CT_DefaultConstraint, CT_OrderExpression, CT_Result, CT_GroupExpression, CT_ParameterValue, CT_ExpressionArgument, Expression

Specifies an expression.

Child Elements:

BitLiteral: A CT_BitLiteral element that specifies a **Boolean** literal.

DateLiteral: A CT_DateLiteral element that specifies a date literal.

DateTimeLiteral: A CT_DateTimeLiteral element that specifies a literal with both date and time components.

DecimalLiteral: A CT_DecimalLiteral element that specifies a decimal literal.

FunctionCall: A CT_FunctionCall element that specifies a function call.

Identifier: A CT_Identifier element that specifies an identifier.

IntegerLiteral: A CT_IntegerLiteral element that specifies an integer literal.

NullLiteral: A CT_NullLiteral element that specifies a null literal.

Original: An Ignored Value (section 2.1.4).

StringLiteral: A CT_StringLiteral element that specifies a string literal.

TimeLiteral: A CT_TimeLiteral element that specifies a time literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ExpressionContents">
  <xsd:sequence>
    <xsd:element name="Original" type="xsd:string" minOccurs="0" wbld:cname="Original"/>
    <xsd:choice>
      <xsd:element name="FunctionCall" type="CT_FunctionCall" wbld:cname="FunctionCall"/>
      <xsd:element name="Identifier" type="CT_Identifier" wbld:cname="Identifier"/>
      <xsd:element name="DecimalLiteral" type="CT_DecimalLiteral"
wbld:cname="DecimalLiteral"/>
      <xsd:element name="IntegerLiteral" type="CT_IntegerLiteral"
wbld:cname="IntegerLiteral"/>
      <xsd:element name="NullLiteral" type="CT_NullLiteral" wbld:cname="NullLiteral"/>
      <xsd:element name="StringLiteral" type="CT_StringLiteral" wbld:cname="StringLiteral"/>
      <xsd:element name="DateTimeLiteral" type="CT_DateTimeLiteral"
wbld:cname="DateTimeLiteral"/>
      <xsd:element name="DateLiteral" type="CT_DateLiteral" wbld:cname="DateLiteral"/>
      <xsd:element name="TimeLiteral" type="CT_TimeLiteral" wbld:cname="TimeLiteral"/>
      <xsd:element name="BitLiteral" type="CT_BitLiteral" wbld:cname="BitLiteral"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.46 CT_FunctionCall

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_NamedExpression, CT_IndexedFunctionCall, CT_ExpressionContents

Specifies a function call in an expression. The child elements specify arguments to the function call. Each of the *n* child elements MUST have an **Index** attribute with a unique value between 0 and *n*-1, inclusive, indicating the child's position in the argument list, with 0 specifying the first argument.

Child Elements:

BitLiteral: A CT_IndexedBitLiteral element that specifies a Boolean literal.

DateLiteral: A CT_IndexedDateLiteral element that specifies a date literal.

DatePartLiteral: A CT_IndexedDatePartLiteral element that specifies a date part literal.

DateTimeLiteral: A CT_IndexedDateTimeLiteral element that specifies a literal containing both date and time components.

DecimalLiteral: A CT_IndexedDecimalLiteral element that specifies a decimal literal.

FunctionCall: A CT_IndexedFunctionCall element that specifies a function call.

Identifier: A CT_IndexedIdentifier element that specifies an identifier.

IntegerLiteral: A CT_IndexedIntegerLiteral element that specifies an integer literal.

NullLiteral: A CT_IndexedNullLiteral element that specifies a null literal.

StringLiteral: A CT_IndexedStringLiteral element that specifies a string literal.

TimeLiteral: A CT_IndexedTimeLiteral element that specifies a time literal.

TypeLiteral: A CT_IndexedTypeLiteral element that specifies a type name literal.

Attributes:

Name: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the function. MUST be one of the following values:

Value
+
-
*
/
%
>
<
>=
<=
=
<>
Abs
And
Avg
Between
Cast
Ceiling
CharIndex
Choose
Coalesce

Value
Concat
Count
DateAdd
DateDiff
DateFromParts
DatePart
DateWithTimeFromParts
Day
EOMonth
Exp
Floor
Format
IIf
In
IsNull
Left
Len
Like
Log
Log10
Lower
LTrim
Max
Min
Month
Not
Now
Or
Parse
Pi

Value
Power
Rand
Replace
Replicate
Right
Round
RTrim
Sign
Sqrt
StDev
Stuff
SubString
Sum
TimeFromParts
Today
Try_Cast
Try_Parse
Update
Upper
UserDisplayName
UserEmailAddress
Var
Year

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FunctionCall">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:choice>
      <xsd:element name="FunctionCall" type="CT_IndexedFunctionCall"
wbld:cname="FunctionCall"/>
      <xsd:element name="Identifier" type="CT_IndexedIdentifier" wbld:cname="Identifier"/>
      <xsd:element name="DecimalLiteral" type="CT_IndexedDecimalLiteral"
wbld:cname="DecimalLiteral"/>
```

```
      <xsd:element name="IntegerLiteral" type="CT_IndexedIntegerLiteral"
wbld:cname="IntegerLiteral"/>
      <xsd:element name="NullLiteral" type="CT_IndexedNullLiteral" wbld:cname="NullLiteral"/>
      <xsd:element name="StringLiteral" type="CT_IndexedStringLiteral"
wbld:cname="StringLiteral"/>
      <xsd:element name="TypeLiteral" type="CT_IndexedTypeLiteral" wbld:cname="TypeLiteral"/>
      <xsd:element name="DatePartLiteral" type="CT_IndexedDatePartLiteral"
wbld:cname="DatePartLiteral"/>
      <xsd:element name="DateTimeLiteral" type="CT_IndexedDateTimeLiteral"
wbld:cname="DateTimeLiteral"/>
      <xsd:element name="DateLiteral" type="CT_IndexedDateLiteral" wbld:cname="DateLiteral"/>
      <xsd:element name="TimeLiteral" type="CT_IndexedTimeLiteral" wbld:cname="TimeLiteral"/>
      <xsd:element name="BitLiteral" type="CT_IndexedBitLiteral" wbld:cname="BitLiteral"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="Name" type="xsd:string" use="required" wbld:cname="Name"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.47 CT_Identifier

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedIdentifier, CT_NamedExpression, CT_ExpressionContents

Specifies the name of an identifier in an expression.

Attributes:

Name: A string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the identifier. MUST NOT be empty. If the element containing this attribute is the descendant of a **CT_FormDef** (section 2.2.3.81) element, then the value MUST be the name of a **control**, a user interface macro variable (section 2.1.2.3.1), or a column (1) in the database application. Otherwise, the value MUST be the name of a column (1) in the database application.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Identifier">
  <xsd:attribute name="Name" type="xsd:string" use="required" wbld:cname="Name"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.48 CT_IntegerLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedIntegerLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents an integer in an expression.

Attributes:

Value: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IntegerLiteral">
  <xsd:attribute name="Value" type="xsd:int" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.49 CT_NullLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedNullLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a null value in an expression.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_NullLiteral"/>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.50 CT_StringLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedStringLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a string value in an expression.

Attributes:

Value: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_StringLiteral">
  <xsd:attribute name="Value" type="xsd:string" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.51 CT_DateTimeLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedDateTimeLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a value with both date and time components in an expression.

Attributes:

Value: A **dateTime** ([XMLSCHEMA2] section 3.2.7) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DateTimeLiteral">
  <xsd:attribute name="Value" type="xsd:dateTime" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.52 CT_TimeLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedTimeLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a time value in an expression.

Attributes:

Value: A **time** ([XMLSCHEMA2] section 3.2.8) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TimeLiteral">
  <xsd:attribute name="Value" type="xsd:time" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.53 CT_DateLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedDateLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a date value in an expression.

Attributes:

Value: A **date** ([XMLSCHEMA2] section 3.2.9) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DateLiteral">
  <xsd:attribute name="Value" type="xsd:date" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.54 CT_BitLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedBitLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a Boolean value in an expression.

Attributes:

Value: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_BitLiteral">
  <xsd:attribute name="Value" type="xsd:boolean" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.55 CT_IndexedFunctionCall

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_FunctionCall, CT_IndexedFunctionCall

Specifies a function call that is an argument to a parent function call in an expression. The child elements specify arguments to this function call. Each of the *n* child elements MUST have an **Index** attribute with a unique value between 0 and *n*-1, inclusive, indicating the child's position in the argument list, with 0 specifying the first argument.

Child Elements:

BitLiteral: A CT_IndexedBitLiteral element that specifies a Boolean literal.

DateLiteral: A CT_IndexedDateLiteral element that specifies a date literal.

DatePartLiteral: A CT_IndexedDatePartLiteral element that specifies a date part literal.

DateTimeLiteral: A CT_IndexedDateTimeLiteral element that specifies a literal containing both date and time components.

DecimalLiteral: A CT_IndexedDecimalLiteral element that specifies a decimal literal.

FunctionCall: A CT_IndexedFunctionCall element that specifies a function call.

Identifier: A CT_IndexedIdentifier element that specifies an identifier.

IntegerLiteral: A CT_IndexedIntegerLiteral element that specifies an integer literal.

NullLiteral: A CT_IndexedNullLiteral element that specifies a null literal.

StringLiteral: A CT_IndexedStringLiteral element that specifies a string literal.

TimeLiteral: A CT_IndexedTimeLiteral element that specifies a time literal.

TypeLiteral: A CT_IndexedTypeLiteral element that specifies a type name literal.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of this function call in the list of arguments of the parent function call.

Name: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the function. MUST be one of the following values:

Value
+
-
*
/
%
>
<
>=
<=
=
<>
Abs
And
Avg
Between
Cast
Ceiling
CharIndex
Choose
Coalesce
Concat
Count
DateAdd
DateDiff
DateFromParts

Value
DatePart
DateWithTimeFromParts
Day
EOMonth
Exp
Floor
Format
IIf
In
IsNull
Left
Len
Like
Log
Log10
Lower
LTrim
Max
Min
Month
Not
Now
Or
Parse
Pi
Power
Rand
Replace
Replicate
Right

Value
Round
RTrim
Sign
Sqrt
StDev
Stuff
SubString
Sum
TimeFromParts
Today
Try_Cast
Try_Parse
Update
Upper
UserDisplayName
UserEmailAddress
Var
Year

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedFunctionCall">
  <xsd:complexContent>
    <xsd:extension base="CT_FunctionCall">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.56 CT_IndexedIdentifier

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies an identifier that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the identifier in the list of arguments.

Name: A string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the identifier. MUST NOT be empty. If the element containing this attribute is the descendant of a **CT_FormDef** (section 2.2.3.81) element, then the value MUST be the name of a control, a user interface macro variable (section 2.1.2.3.1), or a column (1) in the database application. Otherwise, the value MUST be the name of a column (1) in the database application.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedIdentifier">
  <xsd:complexContent>
    <xsd:extension base="CT_Identifier">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.57 CT_IndexedIntegerLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents an integer that is an argument to a function call in an expression.

Attributes:

Index: An int ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedIntegerLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_IntegerLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.58 CT_IndexedNullLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a null value that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedNullLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_NullLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.59 CT_IndexedStringLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a string that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedStringLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_StringLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.60 CT_IndexedDatePartLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a date part in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: An **ST_DatePartLiteral** (section 2.2.4.15) attribute that specifies the date part.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedDatePartLiteral">
  <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
  <xsd:attribute name="Value" type="ST_DatePartLiteral" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.61 CT_IndexedTypeLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a **data type** name literal that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: An **ST_TypeLiteral** (section 2.2.4.16) attribute that specifies the name of the data type.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedTypeLiteral">
  <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
  <xsd:attribute name="Value" type="ST_TypeLiteral" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.62 CT_IndexedDateTimeLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a date with time that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: A **dateTime** ([XMLSCHEMA2] section 3.2.7) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedDateTimeLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_DateTimeLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.63 CT_IndexedTimeLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a time that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: A **time** ([XMLSCHEMA2] section 3.2.8) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedTimeLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_TimeLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.64 CT_IndexedDateLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a date that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: A **date** ([XMLSCHEMA2] section 3.2.9) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedDateLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_DateLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.65 CT_IndexedBitLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a Boolean value that is an argument to a function call in an expression.

Attributes:

Index: An **int** ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedBitLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_BitLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.66 CT_Index

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Unique, Index

Specifies an **index** (section 2.1.1.3) on a table.

Child Elements:

PropertyRef: A **CT_PropertyRefIndex** element (section 2.2.3.88) that specifies the name of a column (1) that is part of the index.

Attributes:

Caption: An **ST_LongString** attribute (section 2.2.4.4) that specifies a caption for the index.

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the index.

ObjectId: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an identifier for the index.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Index">
  <xsd:sequence>
    <xsd:element name="PropertyRef" type="CT_PropertyRefIndex" wbld:cname="PropertyRef"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"
form="qualified"/>
  <xsd:attribute name="ObjectId" wbld:cname="ObjectId" type="ST_ObjectName"
form="qualified"/>
  <xsd:attribute name="Caption" type="ST_LongString" wbld:cname="Caption" form="qualified"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.67 CT_Unique

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: Unique

Specifies a unique constraint (section 2.1.1.4) on a table.

Child Elements:

PropertyRef: A **CT_PropertyRefIndex** element (section 2.2.3.88) that specifies the name of a column (1) that is part of the index.

Attributes:

Caption: An **ST_LongString** attribute (section 2.2.4.4) that specifies a caption for the index.

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the index.

ObjectId: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an identifier for the index.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Unique">
  <xsd:complexContent>
    <xsd:extension base="CT_Index"/>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.68 CT_ConstraintWithExpression

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_CheckConstraint, CT_DefaultConstraint

Specifies a constraint on a table or on a column (1). MUST NOT have a **PropertyRef** child element for a table-wide constraint. MUST have one **PropertyRef** child element if the constraint is associated with a column (1).

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies an expression.

PropertyRef: A **CT_PropertyRef** element (section 2.2.3.87) that specifies the name of the column (1) with which this constraint is associated.

Attributes:

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the constraint.

ObjectId: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an identifier for the constraint.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ConstraintWithExpression">
  <xsd:sequence>
    <xsd:element name="PropertyRef" wbld:cname="PropertyRef" type="CT_PropertyRef"
minOccurs="0"/>
    <xsd:element name="Expression" wbld:cname="Expression" type="CT_ExpressionContents"/>
  </xsd:sequence>
  <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"
form="qualified"/>
  <xsd:attribute name="ObjectId" wbld:cname="ObjectId" type="ST_ObjectName"
form="qualified"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.69 CT_CheckConstraint

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CheckConstraint

Specifies a check constraint (section 2.1.1.1) on a table or on a single column.

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies an expression.

PropertyRef: A **CT_PropertyRef** element (section 2.2.3.87) that specifies the name of the column (1) with which this constraint is associated.

Attributes:

CheckData: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether to check existing data in the column or table for constraint violations.

Message: An **ST_ShortString** (section 2.2.4.3) attribute that specifies the message associated with a constraint violation.

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the constraint.

ObjectId: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an identifier for the constraint.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_CheckConstraint">
  <xsd:complexContent>
    <xsd:extension base="CT_ConstraintWithExpression">
      <xsd:attribute name="CheckData" wbld:cname="CheckData" type="xsd:boolean"
use="optional" form="qualified"/>
      <xsd:attribute name="Message" wbld:cname="Message" type="ST_ShortString" use="optional"
form="qualified"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.70 CT_DefaultConstraint

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: DefaultConstraint

Specifies a default constraint (section 2.1.1.2) on a column (1) in a table.

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies the default value.

PropertyRef: A **CT_PropertyRef** element (section 2.2.3.87) that specifies the name of the column (1).

Attributes:

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the constraint.

ObjectId: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies an identifier for the constraint.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DefaultConstraint">
  <xsd:complexContent>
```

```
    <xsd:restriction base="CT_ConstraintWithExpression">
      <xsd:sequence>
        <xsd:element name="PropertyRef" wbld:cname="PropertyRef" type="CT_PropertyRef"/>
        <xsd:element name="Expression" wbld:cname="Expression" type="CT_ExpressionContents"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.71 CT_OrderExpression

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Ordering, CT_AdHocOrdering

Specifies a single expression and a direction to order.

Child Elements:

Expression: A **CT_ExpressionContents** element (section 2.2.3.45) that specifies a single expression by which to order a result set.

If this element is used inside the **CT_Ordering** element (section 2.2.3.3), the following rules apply. If the expression is not included in the **CT_Results** element (section 2.2.3.9), the columns (1) used in this expression MUST be included in the **CT_Results**. If the **CT_Result** element (section 2.2.3.8) specifying the column (1) contains an **Alias** value, the **CT_Identifier** element (section 2.2.3.47) specifying this column (1) MAY have that value. If the **CT_Result** specifying the column (1) contains **Source** and **Name** values, the **CT_Identifier** specifying this column (1) MAY have the **Source.Name** value. If the **CT_Result** specifying the column (1) contains a **Name** value and if it resolves to a unique **CT_Result**, the **CT_Identifier** specifying this column (1) MAY have that value.

Attributes:

Direction: An **ST_SortDirection** attribute (section 2.2.4.7) that specifies the direction of the order.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_OrderExpression">
  <xsd:all>
    <xsd:element name="Expression" type="CT_ExpressionContents" minOccurs="1"
wbld:cname="Expression"/>
  </xsd:all>
  <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.72 CT_Group

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Groups

Specifies a single column (1) from the input sources on which to group the **CT_Results** (section 2.2.3.9). The query results will have one group for each unique value in the data of that column (1).

Attributes:

Name: An **ST_LongString** (section 2.2.4.4) attribute that specifies the name of one column (1) from the input source specified by the **Source** attribute.

Source: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name or alias of one of the input sources to the query. The input source MUST be listed in the **CT_References** (section 2.2.3.7) collection of the query. If the **CT_Reference** (section 2.2.3.6) element specifying the input source contains an **Alias** value, then this attribute MUST be that value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Group">
  <xsd:attribute name="Source" type="ST_ObjectName" wbld:cname="Source" use="required"/>
  <xsd:attribute name="Name" type="ST_LongString" wbld:cname="Name" use="required"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.73 CT_GroupExpression

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Groups

Specifies a single expression that is used to group the **CT_Results** (section 2.2.3.9) of a Query (section 2.1.3). The expression will be evaluated once for each row (1) in the query results before this grouping is applied, and the query results will then be given a group for each unique result of the expression evaluation.

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies a single expression by which to group the **CT_Results** (section 2.2.3.9) of a Query (section 2.1.3).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_GroupExpression">
  <xsd:all>
    <xsd:element name="Expression" type="CT_ExpressionContents" minOccurs="1"
wbld:cname="Expression"/>
  </xsd:all>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.74 CT_Groups

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Container type for all of the groupings specified in a **CT_Query** (section 2.2.3.18). MUST contain at least one **Group** or **GroupExpression** element. Describes the groups into which records will be placed in the result set. Query results contain one row (1) for each group. If this element is present, then each column (1) that is referenced in a **CT_Result** (section 2.2.3.8) element MUST be included either in a **Group** element or in a **CT_ExpressionContents** (section 2.2.3.45) element that represents an aggregate expression (section 2.1.3.1).

Child Elements:

Group: A CT_Group element that specifies a column (1) from the input sources of a **CT_Query** (section 2.2.3.18).

GroupExpression: A CT_GroupExpression element that specifies an expression. Query results are grouped based on that expression.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Groups">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" wbld:cname="Group" type="CT_Group"/>
    <xsd:element name="GroupExpression" wbld:cname="GroupExpression"
type="CT_GroupExpression"/>
  </xsd:choice>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.75 CT_OutputParameter

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_ParameterValues

Specifies one local variable to store an output parameter value.

Attributes:

LocalVarName: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of a local variable which is going to store the output parameter value.

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the output parameter.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_OutputParameter">
  <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
  <xsd:attribute name="LocalVarName" type="ST_ObjectName" use="required"
wbld:cname="LocalVarName"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.76 CT_TopRows

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Specifies a number that limits the number of records returned in the results of a Query (section 2.1.3).

Attributes:

Rows: A **positiveInteger** ([XMLSCHEMA2] section 3.3.25) attribute that specifies the number of records to be returned as results. If the Query (section 2.1.3) doesn't include a **CT_Ordering** (section 2.2.3.3) element, the first **Rows** number of records are returned. If the Query (section 2.1.3) includes a **CT_Ordering** (section 2.2.3.3) element, the first **Rows** number of records, as sorted by that ordering, are returned.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TopRows">
  <xsd:attribute name="Rows" type="xsd:positiveInteger" use="required" wbld:cname="Rows"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.77 CT_TopPercent

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Query

Specifies a percentage that limits the percent of records returned as results of a Query (section 2.1.3).

Attributes:

Percent: An ST_Percent attribute that specifies the number of records to be returned as results, expressed as a percentage. If the Query (section 2.1.3) doesn't include a **CT_Ordering** (section 2.2.3.3) element, the first **Percent** percent of records are returned. If the Query (section 2.1.3) includes a **CT_Ordering** (section 2.2.3.3) element the first **Percent** percent of records, as sorted by that ordering, are returned.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TopPercent">
  <xsd:attribute name="Percent" use="required" wbld:cname="Percent" type="ST_Percent"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.78 CT_ExpressionArgument

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Action

Specifies an argument of a macro action (section 2.2.5.1) with a single expression.

Child Elements:

Expression: A **CT_ExpressionContents** (section 2.2.3.45) element that specifies a single expression as the value of the argument.

Attributes:

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the argument.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ExpressionArgument">
  <xsd:sequence>
    <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
  </xsd:sequence>
  <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.79 CT_Expressions

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_FormDef

Container type for the expressions in a **CT_FormDef** (section 2.2.3.81).

Child Elements:

Expression: A **CT_NamedExpression** (section 2.2.3.86) element that specifies a single expression in a **CT_FormDef** (section 2.2.3.81).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Expressions">
  <xsd:sequence>
    <xsd:element name="Expression" minOccurs="0" maxOccurs="unbounded"
type="CT_NamedExpression" wbld:cname="Expression"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.80 CT_EventDataMacro

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: EventDataMacro

Specifies an Event Data Macro (section 2.1.2.2) on a table.

Child Elements:

DataMacro: A **CT_DataMacro** element (section 2.2.3.44) that specifies the Event Data Macro (section 2.1.2.2). MUST include the **Event** attribute.

Attributes:

Version: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the version number of the Event Data Macro (section 2.1.2.2).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_EventDataMacro">
  <xsd:sequence>
    <xsd:element name="DataMacro" minOccurs="1" maxOccurs="1" type="CT_DataMacro"
wbld:cname="DataMacro"/>
  </xsd:sequence>
  <xsd:attribute name="Version" type="xsd:string" wbld:cname="Version" use="optional"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.81 CT_FormDef

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: FormDef

Specifies a form (1).

Child Elements:

Expressions: A **CT_Expressions** (section 2.2.3.79) element that specifies all the expressions embedded in the form.

Html: A **string** ([XMLSCHEMA2] section 3.2.1) element that specifies the form and its controls, properties and layout. MUST be present. MUST be a **CDATA section** ([XML] section 2.7) that contains valid **HTML** markup.

Query: A **CT_Query** (section 2.2.3.18) element that specifies a Query (section 2.1.3) to serve as the data source (1) of the form.

RecordSource: A **string** ([XMLSCHEMA2] section 3.2.1) element that specifies the name of a Table (section 2.1.1) or Query (section 2.1.3) that is the data source (1) of the form. If the **Query** element is present, **RecordSource** MUST specify the same query.

UserInterfaceMacros: A **CT_UserInterfaceMacros** (section 2.2.3.41) element that specifies all the User Interface Macros embedded in the form.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FormDef">
  <xsd:sequence>
```

```
    <xsd:element name="RecordSource" type="xsd:string" wbld:cname="RecordSource"
minOccurs="0"/>
    <xsd:element name="Html" wbld:cname="Html" type="xsd:string"/>
    <xsd:element name="UserInterfaceMacros" wbld:cname="UserInterfaceMacros"
wbld:codeName="UserInterfaceMacros" type="CT_UserInterfaceMacros" minOccurs="0"/>
    <xsd:element name="Expressions" wbld:cname="Expressions" wbld:codeName="Expressions"
type="CT_Expressions" minOccurs="0"/>
    <xsd:element name="Query" wbld:cname="Query" wbld:codeName="Query" type="CT_Query"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.82 CT_ExtendedAttribute

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_ExtendedAttributes

Specifies the name and the value of a property of a table or column (1).

Attributes:

Name: An **ST_ObjectName** (section 2.2.4.1) attribute that specifies the name of the property.

Value: An **ST_LongString** (section 2.2.4.4) attribute that specifies the value of the property.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ExtendedAttribute" wbld:cname="CT_ExtendedAttribute">
  <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"/>
  <xsd:attribute name="Value" wbld:cname="Value" type="ST_LongString" use="required"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.83 CT_ExtendedAttributes

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: ExtendedAttributes

Specifies a list of properties of a table or column (1) as name/value pairs.

Child Elements:

ExtendedAttribute: A **CT_ExtendedAttribute** (section 2.2.3.82) element that specifies the name and the value of a table or column property.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ExtendedAttributes" wbld:cname="CT_ExtendedAttributes">
  <xsd:sequence>
```

```
    <xsd:element name="ExtendedAttribute" wbld:cname="ExtendedAttribute"
type="CT_ExtendedAttribute" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.84 CT_DecimalLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedDecimalLiteral, CT_NamedExpression, CT_ExpressionContents

Specifies a literal that represents a decimal number in an expression.

Attributes:

Value: An **ST_Decimal** (section 2.2.4.14) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DecimalLiteral">
  <xsd:attribute name="Value" type="ST_Decimal" use="required" wbld:cname="Value"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.85 CT_IndexedDecimalLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedFunctionCall, CT_FunctionCall

Specifies a literal that represents a decimal number that is an argument to a function call in an expression.

Attributes:

Index: An int ([XMLSCHEMA2] section 3.3.17) attribute that specifies the zero-indexed position of the literal in the list of arguments.

Value: An **ST_Decimal** (section 2.2.4.14) attribute that specifies the value of the literal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_IndexedDecimalLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_DecimalLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.86 CT_NamedExpression

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Expressions

Specifies an expression that has a name associated with it. User Interface Macros (section 2.1.2.1) and controls on the form (1) that are descendants of the same **CT_FormDef** (section 2.2.3.81) element as the **CT_NamedExpression** element can reference the expression using the value of the **Name** attribute.

Child Elements:

BitLiteral: A CT_BitLiteral element that specifies a Boolean literal.

DateLiteral: A CT_DateLiteral element that specifies a date literal.

DateTimeLiteral: A CT_DateTimeLiteral element that specifies a literal with both date and time components.

DecimalLiteral: A CT_DecimalLiteral element that specifies a decimal literal.

FunctionCall: A CT_FunctionCall element that specifies a function call.

Identifier: A CT_Identifier element that specifies an identifier.

IntegerLiteral: A CT_IntegerLiteral element that specifies an integer literal.

NullLiteral: A CT_NullLiteral element that specifies a null literal.

Original: An Ignored Value (section 2.1.4).

StringLiteral: A CT_StringLiteral element that specifies a string literal.

TimeLiteral: A CT_TimeLiteral element that specifies a time literal.

Attributes:

Name: A string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the expression.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_NamedExpression">
  <xsd:complexContent>
    <xsd:extension base="CT_ExpressionContents">
      <xsd:attribute name="Name" type="xsd:string" wbld:cname="Name" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.87 CT_PropertyRef

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_PropertyRefIndex, CT_ConstraintWithExpression, CT_CheckConstraint, CT_DefaultConstraint, PropertyRef

Specifies a reference to a column (1) in a table (section 2.1.1).

Attributes:

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the referenced column (1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_PropertyRef">
  <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"/>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.3.88 CT_PropertyRefIndex

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Index, CT_Unique

Specifies a reference to a column (1) that is part of an index (section 2.1.1.3) in a table (section 2.1.1).

Attributes:

Direction: An **ST_SortDirection** attribute (section 2.2.4.7) that specifies the sort order (3) of the column (1) in the index (section 2.1.1.3).

Name: An **ST_ObjectName** attribute (section 2.2.4.1) that specifies the name of the referenced column (1).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_PropertyRefIndex">
  <xsd:complexContent>
    <xsd:extension base="CT_PropertyRef">
      <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4 Simple Types

2.2.4.1 ST_ObjectName

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: BoundColumn, DisplayColumn, DisplayName, InternalName, ObjectId, ReferencedTable, CT_Index, CT_PropertyRefIndex, CT_PropertyRef, CT_Unique, CT_ConstraintWithExpression, CT_CheckConstraint, CT_DefaultConstraint, CT_ExtendedAttribute, CT_Order, CT_Reference, CT_Parameter, CT_Result, CT_Join, CT_Group, CT_ParameterDefinition, CT_ParameterValue, CT_OutputParameter, CT_Query, CT_Argument, CT_ExpressionArgument, CT_Action, CT_ForEachRecordData, CT_LookupRecordData, CT_CreateRecordData, CT_EditRecordData, CT_UserInterfaceMacro

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies an attribute of an object in a database application. MUST NOT begin with a space character or the '=' character. MUST NOT contain the tab character or any of the following characters: /, \, :, *, ?, ", <, >, ., |, #, {, }, %, ~, &, ;, !, `, [,], or any character whose hexadecimal value is less than 0x20. The length of the string MUST NOT exceed 64 characters.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_ObjectName">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="64"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.2 ST_DataObjectEvent

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_DataMacro

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the type of event that causes a Data Macro (section 2.1.2.2) to be called.

MUST be one of the following values.

Value	Meaning
AfterInsert	The macro is called after a new record has been added.
AfterUpdate	The macro is called after an existing record has been updated.
AfterDelete	The macro is called after an existing record has been removed.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_DataObjectEvent">
```

```
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
      <xsd:enumeration value="AfterInsert" wbld:cname="AfterInsert"/>
      <xsd:enumeration value="AfterUpdate" wbld:cname="AfterUpdate"/>
      <xsd:enumeration value="AfterDelete" wbld:cname="AfterDelete"/>
    </xsd:restriction>
  </xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.3 ST_ShortString

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_CheckConstraint

A **string** ([XMLSCHEMA2] section 3.2.1) that MUST be less than or equal to 255 characters in length.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
  <xsd:simpleType name="ST_ShortString">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.4 ST_LongString

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: Caption, CT_Index, CT_Unique, CT_ExtendedAttribute, CT_Order, CT_AdHocOrder, CT_Result, CT_Join, CT_Group, CT_ParameterDefinition, CT_Argument, CT_DataMacroStatements, CT_DataMacroStatementGroup, CT_UserInterfaceMacroStatements, CT_UserInterfaceMacroStatementGroup

A **string** ([XMLSCHEMA2] section 3.2.1) that MUST be less than or equal to 1024 characters in length.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
  <xsd:simpleType name="ST_LongString">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="1024"/>
    </xsd:restriction>
  </xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.5 ST_JoinDirection

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Join

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the direction of a **CT_Join** (section 2.2.3.10) in a **CT_Query** (section 2.2.3.18). MUST be one of the following values.

Value	Meaning
Inner	The join results include data from rows (1) in the left input source and the right input source where the **LeftProperty** and **RightProperty** values are equal.
Left Outer	The join results include data from all rows (1) in the left input source along with data from rows (1) in the right input source where the **LeftProperty** and **RightProperty** values are equal.
Right Outer	The join results include data from all rows (1) in the right input source along with data from rows (1) in the left input source where the **LeftProperty** and **RightProperty** values are equal.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_JoinDirection">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Inner" wbld:cname="Inner"/>
    <xsd:enumeration value="Left Outer" wbld:cname="LeftOuter"/>
    <xsd:enumeration value="Right Outer" wbld:cname="RightOuter"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.6 ST_FieldType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: UnderlyingType, CT_Parameter, CT_ParameterDefinition

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a data type. MUST be one of the following values.

Value	Meaning
Text	A string value up to 4000 characters in length.
Number	A **floating-point number**.
Yes/No	A Boolean value.
Currency	A monetary value.
Date/Time	A date value that includes time.
LongText	A string value up to the maximum storage size specified by the **nvarchar** ([MSDN-TSQL-Ref]) data type.

Value	Meaning
Date	A date value.
Time	A time value.
Integer	An integer number value.
Decimal	A number value with specified scale and precision, as specified by the **decimal** data type ([MSDN-TSQL-Ref]).

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_FieldType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Text" wbld:cname="Text"/>
    <xsd:enumeration value="Number" wbld:cname="Number"/>
    <xsd:enumeration value="Yes/No" wbld:cname="YesNo"/>
    <xsd:enumeration value="Currency" wbld:cname="Currency"/>
    <xsd:enumeration value="Date/Time" wbld:cname="DateTime"/>
    <xsd:enumeration value="LongText" wbld:cname="LongText"/>
    <xsd:enumeration value="Date" wbld:cname="Date"/>
    <xsd:enumeration value="Time" wbld:cname="Time"/>
    <xsd:enumeration value="Integer" wbld:cname="Integer"/>
    <xsd:enumeration value="Decimal" wbld:cname="Decimal"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.7 ST_SortDirection

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: Direction, CT_PropertyRefIndex, CT_Order, CT_OrderExpression, CT_AdHocOrder

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the direction of an order. MUST be one of the following values.

Value	Meaning
Ascending	The ordering is smallest to largest.
Descending	The ordering is largest to smallest.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_SortDirection">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Ascending" wbld:cname="Ascending"/>
    <xsd:enumeration value="Descending" wbld:cname="Descending"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.8 ST_TextType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: TextType

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the layout or format of a text column (1). MUST be one of the following values.

Value	Meaning
SingleLine	A single line of text.
MultipleLines	Multiple lines of text.
Hyperlink	A hyperlink.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_TextType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SingleLine" wbld:cname="SingleLine"/>
    <xsd:enumeration value="MultipleLines" wbld:cname="MultipleLines"/>
    <xsd:enumeration value="Hyperlink" wbld:cname="Hyperlink"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.9 TStoreGeneratedPattern

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: StoreGeneratedPattern

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the type of a column (1) based on the pattern used to generate its value. MUST be one of the following values.

Value	Meaning
None	The column does not have an auto-generated pattern for its values.
Identity	The values of the column are generated such that each record has a unique value in the column.
Computed	The values of the column are generated by evaluating an expression.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="TStoreGeneratedPattern" wbld:cname="TStoreGeneratedPattern">
  <xsd:restriction base="xsd:token">
```

```
    <xsd:enumeration value="None" wbld:cname="None"/>
    <xsd:enumeration value="Identity" wbld:cname="Identity"/>
    <xsd:enumeration value="Computed" wbld:cname="Computed"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.10 ST_LookupType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: LookupType

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the source of the data in a lookup field. MUST be one of the following values.

Value	Meaning
TableOrQuery	The source is a table or a query.
ValueList	The source is a given list of values.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_LookupType" wbld:cname="ST_LookupType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="TableOrQuery" wbld:cname="TableOrQuery"/>
    <xsd:enumeration value="ValueList" wbld:cname="ValueList"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.11 ST_Percent

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_TopPercent

A **float** ([XMLSCHEMA2] section 3.2.4) that specifies a percentage. MUST be greater than or equal to 0 and less than or equal to 100.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_Percent">
  <xsd:restriction base="xsd:float">
    <xsd:minExclusive value="0"/>
    <xsd:maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.12 ST_UserInterfaceObjectEvent

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_UserInterfaceMacro

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the type of event that causes a macro (section 2.1.2) to be called.

The value MUST be one of the events that is supported by the form or by the type of control that is specified by the **For** attribute of the **CT_UserInterfaceMacro** (section 2.2.3.43) element.

The following table lists the valid events for form and control types. Any control types not listed have no valid events.

Control type	Valid events
Form	OnLoad, OnCurrent
Text Box / Multiline Textbox	OnClick, AfterUpdate
Button	OnClick
Combo Box	AfterUpdate
Label	OnClick
Image	OnClick
Autocomplete Control	OnClick, AfterUpdate
Check Box	AfterUpdate
Hyperlink Control	OnClick, AfterUpdate

The following table lists the meaning of each event.

Value	Meaning
OnClick	Call the macro after a control is clicked.
onclick	Call the macro after a control is clicked.
AfterUpdate	Call the macro after the content of a control is updated.
afterupdate	Call the macro after the content of a control is updated.
OnCurrent	Call the macro after the current record is changed.
oncurrent	Call the macro after the current record is changed.
OnLoad	Call the macro after a Form is loaded.
onload	Call the macro after a Form is loaded.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_UserInterfaceObjectEvent">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="64"/>
    <xsd:enumeration value="OnClick" wbld:cname="OnClick"/>
    <xsd:enumeration value="onclick" wbld:cname="onclick"/>
    <xsd:enumeration value="AfterUpdate" wbld:cname="AfterUpdate"/>
    <xsd:enumeration value="afterupdate" wbld:cname="afterupdate"/>
    <xsd:enumeration value="OnCurrent" wbld:cname="OnCurrent"/>
    <xsd:enumeration value="oncurrent" wbld:cname="oncurrent"/>
    <xsd:enumeration value="OnLoad" wbld:cname="OnLoad"/>
    <xsd:enumeration value="onload" wbld:cname="onload"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.13 ST_Format

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: Format

Specifies the format type for displaying the data in a column (1). The actual formatting is implementation-specific.

Value	Meaning
General Number	Column data is a number formatted with the most compact of either fixed-point or scientific notation.
Currency	Column data is a currency value formatted with a currency symbol.
Fixed	Column data is a number, formatted with integral and decimal digits with optional negative sign.
Standard	Column data is a number formatted with integral and decimal digits, group separators, and a decimal separator with optional negative sign.
Percent	Column data is a number formatted as the number multiplied by 100 and displayed with a percent symbol.
General Date	Column data is a date that optionally includes a time.
Long Date	Column data is a long-form date.
Short Date	Column data is a short-form date.
Long Time	Column data is a long-form time.
Short Time	Column data is a short-form time.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_Format" wbld:cname="ST_Format">
  <xsd:restriction base="xsd:string">
```

```
      <xsd:enumeration value="General Number" wbld:cname="GeneralNumber"/>
      <xsd:enumeration value="Currency" wbld:cname="Currency"/>
      <xsd:enumeration value="Fixed" wbld:cname="Fixed"/>
      <xsd:enumeration value="Standard" wbld:cname="Standard"/>
      <xsd:enumeration value="Percent" wbld:cname="Percent"/>
      <xsd:enumeration value="General Date" wbld:cname="GeneralDate"/>
      <xsd:enumeration value="Long Date" wbld:cname="LongDate"/>
      <xsd:enumeration value="Short Date" wbld:cname="ShortDate"/>
      <xsd:enumeration value="Long Time" wbld:cname="LongTime"/>
      <xsd:enumeration value="Short Time" wbld:cname="ShortTime"/>
    </xsd:restriction>
  </xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.14 ST_Decimal

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedDecimalLiteral, CT_DecimalLiteral

A **decimal** ([XMLSCHEMA2] section 3.2.3) that specifies a number.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_Decimal">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits value="28"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.15 ST_DatePartLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedDatePartLiteral

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a date part literal in an expression. MUST be one of the following values.

Value	Meaning
YEAR	Specifies the Year date part.
QUARTER	Specifies the Quarter date part.
MONTH	Specifies the Month date part.
DAYOFYEAR	Specifies the DayOfYear date part.
DAY	Specifies the Day date part.
WEEK	Specifies the Week date part.

Value	Meaning
WEEKDAY	Specifies the Weekday date part.
HOUR	Specifies the Hour date part.
MINUTE	Specifies the Minute date part.
SECOND	Specifies the Second date part.
MILLISECOND	Specifies the Millisecond date part.
ISO_WEEK	Specifies the ISO_Week date part.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_DatePartLiteral">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="YEAR" wbld:cname="YEAR"/>
    <xsd:enumeration value="QUARTER" wbld:cname="QUARTER"/>
    <xsd:enumeration value="MONTH" wbld:cname="MONTH"/>
    <xsd:enumeration value="DAYOFYEAR" wbld:cname="DAYOFYEAR"/>
    <xsd:enumeration value="DAY" wbld:cname="DAY"/>
    <xsd:enumeration value="WEEK" wbld:cname="WEEK"/>
    <xsd:enumeration value="WEEKDAY" wbld:cname="WEEKDAY"/>
    <xsd:enumeration value="HOUR" wbld:cname="HOUR"/>
    <xsd:enumeration value="MINUTE" wbld:cname="MINUTE"/>
    <xsd:enumeration value="SECOND" wbld:cname="SECOND"/>
    <xsd:enumeration value="MILLISECOND" wbld:cname="MILLISECOND"/>
    <xsd:enumeration value="ISO_WEEK" wbld:cname="ISO_WEEK"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.16 ST_TypeLiteral

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_IndexedTypeLiteral

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a data type. It MUST be one of the following values.

Value	Meaning
FLOAT	Specifies the Float data type.
INTEGER	Specifies the Integer data type.
CURRENCY	Specifies the Currency data type.
YESNO	Specifies the Yes/No data type.
TEXT	Specifies the Text data type.

Value	Meaning
SHORTTEXT	Specifies the ShortText data type.
LONGTEXT	Specifies the LongText data type.
DATEWITHTIME	Specifies the DateWithTime data type.
DATE	Specifies the Date data type.
TIME	Specifies the Time data type.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_TypeLiteral">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="FLOAT" wbld:cname="FLOAT"/>
    <xsd:enumeration value="INTEGER" wbld:cname="INTEGER"/>
    <xsd:enumeration value="CURRENCY" wbld:cname="CURRENCY"/>
    <xsd:enumeration value="YESNO" wbld:cname="YESNO"/>
    <xsd:enumeration value="TEXT" wbld:cname="TEXT"/>
    <xsd:enumeration value="SHORTTEXT" wbld:cname="SHORTTEXT"/>
    <xsd:enumeration value="LONGTEXT" wbld:cname="LONGTEXT"/>
    <xsd:enumeration value="DATEWITHTIME" wbld:cname="DATEWITHTIME"/>
    <xsd:enumeration value="DATE" wbld:cname="DATE"/>
    <xsd:enumeration value="TIME" wbld:cname="TIME"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.17 ST_DecimalPlaces

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: DecimalPlaces

Specifies the number of decimal places used to display a numeric value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_DecimalPlaces">
  <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="15"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.4.18 ST_QuerySourceType

Target namespace: http://schemas.microsoft.com/office/accessservices/2010/12/application

Referenced by: CT_Reference

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the type of the **CT_Reference** (section 2.2.3.6).

Value	Meaning
Table	Input source is a table.
Query	Input source is a query.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_QuerySourceType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Table" wbld:cname="Table"/>
    <xsd:enumeration value="Query" wbld:cname="Query"/>
  </xsd:restriction>
</xsd:simpleType>
```

See section 5.1 for the full W3C XML Schema ([XMLSCHEMA1] section 2.1).

2.2.5 Macros

This section specifies the valid values for the **Action** and **Argument** elements of a macro (section 2.1.2).

2.2.5.1 Actions

Actions are elements of type **CT_Action** (section 2.2.3.20). Actions MUST have a **Name** attribute set to the title of one of the following subsections. These actions can be present in either a Data Macro (section 2.1.2.2) or a User Interface Macro (section 2.1.2.1) unless otherwise specified. Child arguments not specified by an action MUST NOT be present in that action. Child arguments specified by a particular action are optional unless otherwise specified.

2.2.5.1.1 ChangeView

Changes the focus from the active form (1) object to a different form (1) object. Parameters to the form, if any, are specified by the **Parameters** element in **CT_Action** (section 2.2.3.20).

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type can have the following child arguments.

Argument	Meaning
Table (section 2.2.5.2.5)	The name of the table that is a data source (1) of the target form. MUST be present.
View (section 2.2.5.2.6)	The name of the target form (1). MUST be present.
Where (section 2.2.5.2.15)	An **Expression** (section 2.2.1.10) used to select the set of records that will be displayed from the data source (1) of the target form (1).

Argument	Meaning
OrderBy (section 2.2.5.2.11)	The ordering used to sort the set of records from the data source (1) of the target form (1).

2.2.5.1.2 CancelRecordChange

Exits the **CT_CreateRecord** (section 2.2.3.26) or **CT_EditRecord** (section 2.2.3.28) block with no record change.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) element unless that element is contained by a **CT_EditRecord** (section 2.2.3.28) or **CT_CreateRecord** (section 2.2.3.26) element. MUST NOT be present in a User Interface Macro (section 2.1.2.1).

2.2.5.1.3 ClosePopup

Closes the active popup.

MUST NOT be present in a Data Macro (section 2.1.2.2).

2.2.5.1.4 DeleteRecord

If present in a User Interface Macro (section 2.2.1.3), deletes the current record from the data source (1) of the active form (section 2.2.3.81). If present in a Data Macro (section 2.1.2.2), deletes a record based on the argument.

Actions of this type in a Data Macro (section 2.1.2.2) can have the following child argument.

Argument	Meaning
Alias (section 2.2.5.2.1)	The data source (1) from which the record is to be deleted. If this argument is not present, the current record is deleted.

2.2.5.1.5 EditRecord

Sets the current record to edit mode.

MUST NOT be present in a Data Macro (section 2.1.2.2).

2.2.5.1.6 ExitForEachRecord

Exits a running **CT_ForEachRecord** (section 2.2.3.22) loop.

MUST NOT be present in any element except a **CT_DataMacroStatements** (section 2.2.3.38) that is contained by a **CT_ForEachRecord** (section 2.2.3.22) element. MUST NOT be present in a User Interface Macro (section 2.1.2.1).

2.2.5.1.7 GoToControl

Sets the focus to a control on the active form (1) (section 2.2.3.81).

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type can have the following child argument.

Argument	Meaning
ControlName (section 2.2.5.2.2)	The name of the control to which the focus will be set. MUST be present.

2.2.5.1.8 GoToRecord

Sets the current record in the active form (1).

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type can have the following child argument.

Argument	Meaning
Record (section 2.2.5.2.13)	The record that will become current.

2.2.5.1.9 MessageBox

Launches a modal dialog box that displays a message.

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type MUST have the following child argument.

Argument	Meaning
Message (section 2.2.5.2.8)	The message to be displayed in the dialog box. MUST be present.

2.2.5.1.10 OpenPopup

Opens a form (1). Parameters to that form (1) are specified by the **Parameters** element in **CT_Action** (section 2.2.3.20).

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type can have the following child arguments.

Argument	Meaning
View (section 2.2.5.2.6)	The name of the form (1) to open. MUST be present.
Where (section 2.2.5.2.15)	An **Expression** (section 2.2.1.10) that is used to select a set of records from the data source (1) of the target form (1).
OrderBy (section 2.2.5.2.11)	The order used to sort the set of records from the data source (1) of the target form (1).

2.2.5.1.11 RaiseError

Generates an error message.

MUST NOT be present in a User Interface Macro (section 2.1.2.1).

Actions of this type can have the following child argument.

Argument	Meaning
Description (section 2.2.5.2.3)	The description of the error.

2.2.5.1.12 RequeryRecords

Refreshes the data displayed by the active form (1) by querying its data source (1) again.

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type can have the following child arguments.

Argument	Meaning
Where (section 2.2.5.2.15)	An **Expression** (section 2.2.1.10) that is used to select a set of records from the data source (1) of the active form (1).
OrderBy (section 2.2.5.2.11)	The order used to sort the set of records from the data source (1) of the target form (1).

2.2.5.1.13 RunDataMacro

Calls a named Data Macro (section 2.1.2.2). Parameters to the invoked macro are specified by the **Parameters** element in the **CT_Action** (section 2.2.3.20) element that specifies the **RunDataMacro** action.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) element that is contained by a **CT_EditRecord** (section 2.2.3.28) or **CT_CreateRecord** (section 2.2.3.26) element.

Actions of this type can have the following child argument.

Argument	Meaning
MacroName (section 2.2.5.2.7)	The name of the Data Macro (section 2.1.2.2) to be run. MUST be present.

2.2.5.1.14 RunMacro

Runs a named User Interface Macro (section 2.1.2.1).

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type MUST have the following child argument.

Argument	Meaning
MacroName (section 2.2.5.2.7)	The name of the User Interface Macro (section 2.1.2.1) to be run. MUST be present.

2.2.5.1.15 SaveRecord

Saves the current record.

MUST NOT be present in a Data Macro (section 2.1.2.2).

2.2.5.1.16 SetField

Changes the value of a field (3) in the current record.

MUST NOT be present in a **CT_DataMacroStatements** (section 2.2.3.38) element unless that element is contained by a **CT_CreateRecord** (section 2.2.3.26) or **CT_EditRecord** (section 2.2.3.28) element. MUST NOT be present in a User Interface Macro (section 2.1.2.1).

Actions of this type can have the following child arguments.

Argument	Meaning
Field (section 2.2.5.2.4)	The name of the field (3) in the record that will be updated. MUST be present.
Value (section 2.2.5.2.14)	The new value for the target field (3) in the record. MUST be present.

2.2.5.1.17 SetLocalVar

Sets the value of a local variable (section 2.1.2.3.2). Creates the variable if it hasn't been created already.

MUST NOT be present in a User Interface Macro (section 2.1.2.1).

Actions of this type MUST have the following child arguments.

Argument	Meaning
Name (section 2.2.5.2.9)	The name of the local variable (section 2.1.2.3.2). MUST be present.
Value (section 2.2.5.2.14)	The new value of the local variable (section 2.1.2.3.2). MUST be present.

2.2.5.1.18 SetProperty

Sets the value of a property for a control.

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type can have the following child arguments.

Argument	Meaning
ControlName (section 2.2.5.2.2)	The name of the control on which to set the property. If this argument is not present, the property is set for the active object.
Property (section 2.2.5.2.12)	The name of the property to be set. MUST be present.
Value (section 2.2.5.2.14)	The new value of the property.

2.2.5.1.19 SetReturnVar

Sets the value of a return variable (section 2.1.2.3.3). Creates the return variable (section 2.1.2.3.3) if it hasn't been created yet.

MUST NOT be present in a User Interface Macro (section 2.1.2.1).

Actions of this type can have the following child arguments.

Argument	Meaning
Name (section 2.2.5.2.9)	The name of the return variable. MUST be present.
Value (section 2.2.5.2.14)	The new value of the return variable (section 2.1.2.3.3) in a Data Macro (section 2.1.2.2). MUST be present.

2.2.5.1.20 SetVariable

Sets the value of a user interface macro variable (section 2.1.2.3.1). Creates the variable if it hasn't been created yet.

MUST NOT be present in a Data Macro (section 2.1.2.2).

Actions of this type MUST have the following child arguments.

Argument	Meaning
Variable (section 2.2.5.2.10)	The name of the user interface macro variable (section 2.1.2.3.1). MUST be present.
Value (section 2.2.5.2.14)	The new value of the user interface macro variable (section 2.1.2.3.1). MUST be present.

2.2.5.1.21 StopMacro

Stops the currently running macro (section 2.1.2).

2.2.5.2 Arguments

Arguments are elements of type **CT_Argument** (section 2.2.3.19) or **CT_ExpressionArgument** (section 2.2.3.78) that exist as child elements of a **CT_Action** (section 2.2.3.20) element. The **Name** attribute of the element MUST be the title of one of the following subsections.

2.2.5.2.1 Alias

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a data source (1). The value of this argument corresponds to the **Alias** attribute of **CT_ForEachRecordData** (section 2.2.3.21) or **CT_LookupRecordData** (section 2.2.3.23).

Applies to: **DeleteRecord** (section 2.2.5.1.4)

2.2.5.2.2 ControlName

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a control. The value MUST conform to the restrictions specified by ST_ObjectName (section 2.2.4.1).

Applies to: **GoToControl** (section 2.2.5.1.7), **SetProperty** (section 2.2.5.1.18)

2.2.5.2.3 Description

Specifies the description of an error. MUST be a **CT_Argument** (section 2.2.3.19) or a **CT_ExpressionArgument** (section 2.2.3.78).

Applies to: **RaiseError** (section 2.2.5.1.11)

2.2.5.2.4 Field

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a field (3).

Applies to: **SetField** (section 2.2.5.1.16)

2.2.5.2.5 Table

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a table. The value MUST conform to the restrictions specified by **ST_ObjectName** (section 2.2.4.1).

Applies to: **ChangeView** (section 2.2.5.1.1)

2.2.5.2.6 View

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a form (1). The value MUST conform to the restrictions specified by **ST_ObjectName** (section 2.2.4.1).

Applies to: **ChangeView** (section 2.2.5.1.1), **OpenPopup** (section 2.2.5.1.10)

2.2.5.2.7 MacroName

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a Macro. The value MUST conform to the restrictions specified by **ST_ObjectName** (section 2.2.4.1).

Applies to: **RunDataMacro** (section 2.2.5.1.13), **RunMacro** (section 2.2.5.1.14)

2.2.5.2.8 Message

Specifies text that is at most 255 characters in length. MUST be a **CT_Argument** (section 2.2.3.19) or a **CT_ExpressionArgument** (section 2.2.3.78).

Applies to: **MessageBox** (section 2.2.5.1.9)

2.2.5.2.9 Name

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a variable.

Applies to: **SetLocalVar** (section 2.2.5.1.17), **SetReturnVar** (section 2.2.5.1.19)

2.2.5.2.10 Variable

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of a variable.

Applies to: **SetVariable** (section 2.2.5.1.20)

2.2.5.2.11 OrderBy

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a sort order (3). MUST be **XML** that specifies a **CT_AdHocOrdering** (section 2.2.3.5) element.

Applies to: **ChangeView** (section 2.2.5.1.1), **OpenPopup** (section 2.2.5.1.10), **RequeryRecords** (section 2.2.5.1.12)

2.2.5.2.12 Property

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a property on a control. MUST be one of the following values.

Value	Meaning
BackColor	Specifies the background color of a control.
Caption	Specifies the caption attribute of a control.
Enabled	Specifies whether a control is enabled.
ForeColor	Specifies the foreground color of a control.
Value	Specifies the value displayed by a control.
Visible	Specifies whether a control is enabled.

Applies to: **SetProperty** (section 2.2.5.1.18)

2.2.5.2.13 Record

A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a record in the data source (1). MUST be one of the following values.

Value	Meaning
First	The first record in the data source (1).
Last	The last record in the data source (1).
Next	The record following the current record in the data source (1).
Previous	The record prior to the current record in the data source (1).

Applies to: **GoToRecord** (section 2.2.5.1.8)

2.2.5.2.14 Value

Specifies the value to be assigned to a field (3), variable or property.

When present in a **SetVariable** (section 2.2.5.1.20), **SetField** (section 2.2.5.1.16), **SetLocalVar** (section 2.2.5.1.17), or **SetReturnVar** (section 2.2.5.1.19) action, this argument MUST be a **CT_ExpressionArgument** (section 2.2.3.78). When present in a **SetProperty** (section 2.2.5.1.18) action, this argument MUST be a **CT_Argument** (section 2.2.3.19) or a **CT_ExpressionArgument** (section 2.2.3.78).

Applies to: **SetVariable** (section 2.2.5.1.20), **SetField** (section 2.2.5.1.16), **SetLocalVar** (section 2.2.5.1.17), **SetReturnVar** (section 2.2.5.1.19), **SetProperty** (section 2.2.5.1.18)

2.2.5.2.15 Where

A **CT_ExpressionArgument** (section 2.2.3.78) that specifies an expression that is used to select a set of records from the data source (1) of an object.

Applies to: **ChangeView** (section 2.2.5.1.1), **OpenPopup** (section 2.2.5.1.10), **RequeryRecords** (section 2.2.5.1.12)

3 Structure Examples

This section contains valid examples for defining different objects in a data application.

3.1 Tables

The following section contains a complete and valid example of a table (section 2.1.1) in a data application.

```
<Schema Namespace="Access.Store"
    xmlns="http://schemas.microsoft.com/ado/2008/09/edm"
    xmlns:axl="http://schemas.microsoft.com/office/accessservices/2010/12/application"
  >
  <EntityContainer Name="AccessStoreContainer">
    <EntitySet Name="Tasks"
               EntityType="Access.Store.Tasks"/>
    <AssociationSet Name="FK_Tasks_A702819E-F124-4AB6-92C7-CE90E2460A89"
        Association="Access.Store.FK_Tasks_A702819E-F124-4AB6-92C7-CE90E2460A89">
      <End EntitySet="Employees"/>
      <End EntitySet="Tasks"/>
    </AssociationSet>
  </EntityContainer>
  <EntityType Name="Tasks">
    <Key>
      <PropertyRef Name="ID"/>
    </Key>
    <Property Name="ID"
              Type="Int32"
              axl:StoreGeneratedPattern="Identity"
              axl:ObjectId="ID"
              axl:Caption="ID"
              Nullable="false"/>
    <Property Name="TaskTitle"
              Type="String"
              Unicode="true"
              axl:TextType="SingleLine"
              axl:ObjectId="TaskTitle"
              MaxLength="220"
              axl:Caption="Task Title"
              Nullable="false"/>
    <Property Name="Description"
              Type="String"
              Unicode="true"
              axl:TextType="MultipleLines"
              axl:ObjectId="Description"
              MaxLength="Max"
              axl:Caption="Description"/>
    <Property Name="DueDate"
              Type="DateTime"
              axl:UnderlyingType="Date"
              axl:ObjectId="DueDate"
              axl:Format="Short Date"
              axl:Caption="Due Date"/>
    <Property Name="PercentComplete"
              Type="Double"
              axl:ObjectId="PercentComplete"
              axl:Caption="Percent Complete"
              Nullable="false"/>
```

```
    <Property Name="Assigned To"
              Type="Int32"
              axl:ObjectId="Assigned To"
              axl:Description="Each task is assigned to someone. This field creates a
 relationship between the task and the person record that owns it."
              axl:LookupType="TableOrQuery"
              axl:ReferencedTable="Employees"
              axl:BoundColumn="ID"
              axl:DisplayColumn="DisplayNameFirstLast"
              axl:Direction="Ascending"/>
    <axl:Index axl:Name="IX_Tasks_B7A69FCF-4725-49DD-9F66-186800A000BF"
               axl:ObjectId="IX_Tasks_B7A69FCF-4725-49DD-9F66-186800A000BF"
               axl:Caption="Assigned To">
      <axl:PropertyRef Name="Assigned To"
                   Direction="Ascending"/>
    </axl:Index>
    <axl:DefaultConstraint
        axl:Name="DF_Tasks_37091DC8-3C38-4FD5-B24F-68517DCD5535">
      <axl:PropertyRef Name="DueDate"/>
      <axl:Expression>
        <axl:Original>Today()</axl:Original>
        <axl:FunctionCall Name="Today"/>
      </axl:Expression>
    </axl:DefaultConstraint>
    <axl:CheckConstraint axl:Name="CK_Tasks_1948FC46-FB42-4C23-9717-7BD6D587EA7F"
        axl:CheckData="false"
        axl:Message="Due date cannot bet set to a date earlier than today.">
      <axl:PropertyRef Name="DueDate"/>
      <axl:Expression>
        <axl:Original>[DueDate]&amp;gt;=Today()</axl:Original>
        <axl:FunctionCall Name="&amp;gt;=">
          <axl:Identifier Name="DueDate"
                          Index="0"/>
          <axl:FunctionCall Name="Today"
                            Index="1"/>
        </axl:FunctionCall>
      </axl:Expression>
    </axl:CheckConstraint>
    <axl:EventDataMacro>
      <axl:DataMacro Event="AfterInsert"/>
    </axl:EventDataMacro>
    <axl:EventDataMacro>
      <axl:DataMacro Event="AfterUpdate"/>
    </axl:EventDataMacro>
    <axl:EventDataMacro>
      <axl:DataMacro Event="AfterDelete"/>
    </axl:EventDataMacro>
    <axl:ExtendedAttributes>
      <axl:ExtendedAttribute Name="NounID"
                             Value="56d3a7db-f3c4-41fc-912d-f0ce3e4d5cbb"/>
      <axl:ExtendedAttribute Name="SingularName"
                             Value="Task"/>
    </axl:ExtendedAttributes>
  </EntityType>
  <Association Name="FK_Tasks_A702819E-F124-4AB6-92C7-CE90E2460A89"
               axl:ObjectId="FK_Tasks_A702819E-F124-4AB6-92C7-CE90E2460A89">
    <End Type="Access.Store.Employees"
         Role="Employees"
         Multiplicity="1"/>
```

```
      <End Type="Access.Store.Tasks"
           Role="Tasks"
           Multiplicity="*">
        <OnDelete Action="None"/>
      </End>
      <ReferentialConstraint>
        <Principal Role="Employees">
          <PropertyRef Name="ID"/>
        </Principal>
        <Dependent Role="Tasks">
          <PropertyRef Name="Assigned To"/>
        </Dependent>
      </ReferentialConstraint>
    </Association>
  </Schema>
```

This protocol is used in conjunction with the Conceptual Schema Definition File Format ([MC-CSDL]) to describe the properties of the table schema.

The TaskTitle column is defined by the **Property** element with the **Name** attribute value equal to `TaskTitle`. The **axl:TextType** attribute specifies that the text format of the column is a single line of text. The **axl:Caption** attribute defines the label text as `Task Title`.

The DueDate column is defined by the **Property** element with the **Name** attribute value equal to `DueDate`. The **axl:Format** attribute specifies that the date value is formatted as a `Short Date`.

The Assigned To column is a lookup field defined by the **Property** element with the **Name** attribute value equal to `Assigned To`. The **axl:Description** attribute provides descriptive text for the field. The **axl:LookupType** attribute is set to **TableOrQuery**, which means this field is getting its values from another table or query. The **axl:ReferencedTable** attribute specifies that this field is getting its values from the `Employees` table. The **axl:BoundColumn** attribute specifies that the field will store values from the `ID` field in `Employees` table, and the **axl:DisplayColumn** attribute specifies that the field will display values from the `DisplayNameFirstLast` field in the `Employees` table.

The **axl:Index** element specifies that there is an index on the `Assigned To` field in the Tasks table.

The **axl:DefaultConstraint** element specifies the default value of the `DueDate` field. It is an expression using the `Today()` function that returns the current date value.

The **axl:CheckConstraint** element defines the validation rule for the `DueDate` field. The expression is set to `[DueDate] >= Today()`, which validates that the values entered for the field are greater than or equal to the current date.

The **axl:EventDataMacro** elements are all empty, which signifies that the `Tasks` table does not define any Data Macros (section 2.1.2.2).

The **Association** and **AssociationSet** elements specify a relationship between the `Tasks` table and the `Employees` table. For more information about these elements, see [MC-CSDL] section 2.1.8 and [MC-CSDL] section 2.1.18.

3.2 Forms

The following section contains an example of forms in data applications.

```
<FormDef xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">
```

```
  <Html>
    <![CDATA[<?xml version="1.0" encoding="UTF-8"?>
   <html lang="en-us" dir="ltr">
          ...
   </html>]]>
  </Html>
</FormDef>
```

The **Html** element specifies the HTML of the form web page. Note that the majority of the HTML content has been removed from this example for brevity.

3.3 UI Macros

The following subsections contain complete and valid examples of UI Macros in data applications.

3.3.1 Validation and Exceptions

The following example demonstrates the use of a User Interface Macro (section 2.1.2.1) to display detailed customer information from a form that is viewing the orders of multiple customers.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<UserInterfaceMacros
xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">

  <UserInterfaceMacro For="DetailButton" Event="onclick">
    <Statements>
      <Action Name="OpenPopup">
        <Argument Name="View">CustomerDetails</Argument>
        <ExpressionArgument Name="Where">
          <Expression>
            <Original>[ID]=[View].[CustomerIDCombobox]</Original>
            <FunctionCall Name="=">
              <Identifier Name="ID" Index="0"/>
              <Identifier Name="View.CustomerIDCombobox" Index="1"/>
            </FunctionCall>
          </Expression>
        </ExpressionArgument>
      </Action>
    </Statements>
  </UserInterfaceMacro>

</UserInterfaceMacros>
```

This sample contains an embedded User Interface Macro that opens the "CustomerDetails" form in a new window using the **OpenPopup** (section 2.2.5.1.10) action. The "CustomerDetails" form will display a record where the **ID** field matches the value from the "CustomerIDCombobox" control of the current form.

The **UserInterfaceMacro** element (section 2.2.1.5) defines the macro. It is triggered by the "OnClick" event of the "DetailButton" command button of the form that contains it, as specified by the **Event** and **For** attributes, respectively.

3.3.2 Standalone

The following example describes a standalone User Interface Macro (section 2.1.2.1) named ShowCustomerDetails. Its behavior when called is the same as that of the previous example.

```
<UserInterfaceMacro
xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">
  <Statements>
    <Action Name="OpenPopup">
      <Argument Name="View">CustomerDetails</Argument>
      <ExpressionArgument Name="Where">
        <Expression>
          <Original>[ID]=[View].[CustomerIDCombobox]</Original>
          <FunctionCall Name="=">
            <Identifier Name="ID" Index="0"/>
            <Identifier Name="View.CustomerIDCombobox" Index="1"/>
          </FunctionCall>
        </Expression>
      </ExpressionArgument>
    </Action>
  </Statements>
</UserInterfaceMacro>
```

The standalone macro can be called from a command button by specifying the following **UserInterfaceMacros** (section 2.2.1.3) element:

```
<UserInterfaceMacros
xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">
  <UserInterfaceMacro For="DetailButton" Event="onclick">
    <Statements>
      <Action Name="RunMacro">
        <Argument Name="MacroName">ShowCustomerDetails</Argument>
      </Action>
    </Statements>
  </UserInterfaceMacro>
</UserInterfaceMacros>
```

The **Argument** element specifies that the "ShowCustomerDetails" macro is called when the command button named "DetailButton" is clicked.

3.4 Data Macros

The following examples contain complete and valid examples of Data Macros (section 2.1.2.2) in data applications.

3.4.1 Validate Delete

The following example shows a Data Macro (section 2.1.2.2) that prevents the user from deleting a record in the Tasks table if the task has not been finished. A task is finished when the value in the **PercentComplete** field of the table equals 100. The macro also specifies an error to display to the user if the user attempts to delete an unfinished task.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<DataMacros xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">
```

```
  <DataMacro Event="AfterDelete">
    <Statements>
      <ConditionalBlock>
        <If>
          <Condition>
            <Expression>
              <Original>[PercentComplete]&amp;lt;100</Original>
              <FunctionCall Name="&amp;lt;">
                <Identifier Name="PercentComplete" Index="0"/>
                <IntegerLiteral Value="100" Index="1"/>
              </FunctionCall>
            </Expression>
          </Condition>
          <Statements>
            <Action Name="RaiseError">
              <Argument Name="Description">This task cannot be deleted until it has been
finished</Argument>
            </Action>
          </Statements>
        </If>
      </ConditionalBlock>
    </Statements>
  </DataMacro>

</DataMacros>
```

This macro exists in the context of the Tasks table. The **Condition** element specifies an expression, "=[PercentComplete] < 100". If the expression evaluates to true — that is, if the value of the **PercentComplete** field in the row (1) about to be deleted is less than 100 — then the commands in the **Statements** element will be performed. "[PercentComplete]" in this example refers to the field (3) in Tasks. The only command is the **RaiseError** (section 2.2.5.1.11) action, which will generate an application error and display a dialog box to the user with the following message: "This task cannot be deleted until it has been finished".

3.4.2 After Insert

This Data Macro (section 2.1.2.2) example demonstrates how to increment the value of the **CurrentTaskCount** field (3) in the "Users" table after a new record is added and assigned to that user in the "Tasks" table. The **DataMacro** element is a child of a **CT_DataMacros** (section 2.2.3.42).

```
<?xml version="1.0" encoding="UTF-16" standalone="no"?>
<DataMacros xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">

  <DataMacro Event="AfterInsert">
    <Statements>

      <LookUpRecord>
        <Data>
          <Reference>Users</Reference>
          <WhereCondition>
            <Expression>
              <Original>[ID]=[Tasks].[Assigned To]</Original>
              <FunctionCall Name="=">
                <Identifier Name="ID" Index="0"/>
                <Identifier Name="Tasks.Assigned To" Index="1"/>
```

```
            </FunctionCall>
          </Expression>
        </WhereCondition>
      </Data>

      <Statements>
        <EditRecord>
          <Data/>
          <Statements>
            <Action Name="SetField">
              <Argument Name="Field">Users.CurrentTaskCount</Argument>
              <ExpressionArgument Name="Value">
                <Expression>
                  <Original>[Users].[CurrentTaskCount]+1</Original>
                  <FunctionCall Name="+">
                    <Identifier Name="Users.CurrentTaskCount" Index="0"/>
                    <IntegerLiteral Value="1" Index="1"/>
                  </FunctionCall>
                </Expression>
              </ExpressionArgument>
            </Action>
          </Statements>
        </EditRecord>
      </Statements>

    </LookUpRecord>

  </Statements>
 </DataMacro>

</DataMacros>
```

This macro exists in the context of the "Tasks" table. The **Event** attribute of **CT_DataMacro** (section 2.2.3.44) is set to **AfterInsert**, as specified by **ST_DataObjectEvent** (section 2.2.4.2), which means that this macro will be run after a record is added to "Tasks". The **Data** child element of **CT_LookupRecord** (section 2.2.3.24) specifies a set of records from the "Users" table. The **EditRecord** element specifies that the **SetField** action (section 2.2.5.1.16) will be performed on all records specified by the **Data** element, in this case the named context from the **LookupRecord.Data** element. The **Field** (section 2.2.5.2.4) argument specifies that the field (3) to be updated is "Users.CurrentTaskCount", and the **Value** (section 2.2.5.2.14) argument specifies that the new value of the field (3) is its current value plus 1.

3.4.3 Named Macro

This Data Macro (section 2.1.2.2) represents another way of incrementing the "CurrentTaskCount" field of the "Users" list.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<DataMacros xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">
  <DataMacro>
    <Parameters>
      <Parameter Name="UserID" Type="Integer"/>
    </Parameters>
    <Statements>
      <LookUpRecord>
        <Data>
```

```
            <Reference>Users</Reference>
            <WhereCondition>
              <Expression>
                <Original>[Users].[ID]=[UserID]</Original>
                <FunctionCall Name="=">
                  <Identifier Name="Users.ID" Index="0"/>
                  <Identifier Name="UserID" Index="1"/>
                </FunctionCall>
              </Expression>
            </WhereCondition>
          </Data>
          <Statements>
            <EditRecord>
              <Data/>
              <Statements>
                <Action Name="SetField">
                  <Argument Name="Field">Users.CurrentTaskCount</Argument>
                  <ExpressionArgument Name="Value">
                    <Expression>
                      <Original>[Users].[CurrentTaskCount]+1</Original>
                      <FunctionCall Name="+">
                        <Identifier Name="Users.CurrentTaskCount" Index="0"/>
                        <IntegerLiteral Value="1" Index="1"/>
                      </FunctionCall>
                    </Expression>
                  </ExpressionArgument>
                </Action>
              </Statements>
            </EditRecord>
          </Statements>
        </LookUpRecord>
      </Statements>
    </DataMacro>
  </DataMacros>
```

The first **DataMacro** element is designed to be called from other macros. Its behavior when called is the same as that of the previous example. It also specifies a **Parameters** element (section 2.2.3.12), with the **Name** attribute of `UserID`. The value of the parameter can then be used by referencing the parameter name, as demonstrated in the **Data.WhereCondition** element, which sets a filter on the records retrieved by the **Data** element to include only those where the value of `Users.ID` is equal to the value of the parameter passed in by the calling macro.

The following **DataMacro** element, which is associated with `Tasks`, is an example of how to call a Named Data Macro (section 2.1.2.2). Its **Event** attribute is set to **AfterInsert**, which is called after a new record is added to the `Tasks` table.

```
  <DataMacro Event="AfterInsert">
      <Statements>
        <Action Name="RunDataMacro">
          <Argument Name="MacroName">IncrementTaskCount</Argument>
          <Parameters>
            <Parameter Name="UserID">
              <Expression>
                <Original>[Assigned To]</Original>
                <Identifier Name="Assigned To"/>
              </Expression>
            </Parameter>
```

```
        </Parameters>
      </Action>
    </Statements>
  </DataMacro>
```

The **RunDataMacro** (section 2.2.5.1.13) action takes the **MacroName** (section 2.2.5.2.7) argument, the value of which is `IncrementTaskCount`. The **Parameter** element specifies an **Expression** (section 2.2.1.10) element, which evaluates to the value of the `AssignedTo` field (3) in the newly added record in `Tasks`. For example, if the newly added record were assigned to a user with an `ID` of `5`, then the `IncrementTaskCount` Data Macro (section 2.1.2.2) would be called with a value of `5` for the `UserID` parameter, and the **WhereCondition** element of the macro would evaluate to `[ID]=5`.

3.5 Queries

The following examples contain complete and valid examples of Queries (section 2.1.3).

3.5.1 Single Reference Query

The following example shows a Query (section 2.1.3) that references a single table called Issues. The purpose of this query is to return all **Summary**, **Status** and **DueDate** fields of the records from the Issues table, containing unique selected field values, in descending order of the **DueDate** field, that meet the following criteria:

1. The issue is not closed.

2. The priority value is less than 3.

SQL:

```
SELECT DISTINCT [Issues].[Summary], [Issues].[Status], [Issues].[DueDate]
FROM [dbo].[Issues]
WHERE [Issues].[Status] <> N'Closed' AND [Issues].[Priority] < N'3'
```

Query AXL:

```
<?xml version="1.0" encoding="utf-8"?>
<Query Distinct="true"
xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">

  <References>
    <Reference Source="Issues"></Reference>
  </References>

  <Results>
    <Property Source="Issues" Name="Summary"></Property>
    <Property Source="Issues" Name="Status"></Property>
    <Property Source="Issues" Name="DueDate"></Property>
  </Results>

  <Restriction>
    <Expression>
      <Original>(((Issues.Status)&amp;lt;&amp;gt;"Closed") AND ((Issues.Priority)&amp;lt;"3"))</Original>
      <FunctionCall Name="And">
        <FunctionCall Name="&amp;lt;&amp;gt;" Index="0">
```

```
          <Identifier Name="Issues.Status" Index="0"></Identifier>
          <StringLiteral Value="Closed" Index="1"></StringLiteral>
        </FunctionCall>
        <FunctionCall Name="&lt;" Index="1">
          <Identifier Name="Issues.Priority" Index="0"></Identifier>
          <StringLiteral Value="3" Index="1"></StringLiteral>
        </FunctionCall>
      </FunctionCall>
    </Expression>
  </Restriction>

  <Ordering>
    <Order Direction="Descending" Source="Issues" Name="DueDate"></Order>
  </Ordering>
</Query>
```

The **Distinct** attribute in the **Query** element specifies that the query will return only unique values. If the attribute were "false" or not present, the query would return all values.

The **References** element specifies that the "Issues" table is the data source (1) for this query.

The **Results** element specifies that the query will return the fields "Summary", "Status" and "DueDate" from the "Issues" table.

The **Restriction** element specifies an expression that selects only those records where the values of the "Status" field do not equal "Closed" and the "Priority" field of the issue is less than 3.

The **Ordering** element specifies that the results are sorted in descending order based on the "DueDate" field.

3.5.2 Multiple Reference Query with Join

The following example is a Query (section 2.1.3) that joins two tables to produce a result.

The purpose of this query is to return the total number of issues for each customer. The query references two tables, `Customers` and `Issues`. It performs a left outer join between the `Customers` table on the left and `Issues` table on the right. The result set is grouped by the `DisplayName` field from the `Customers` table and the number of issues is returned.

SQL:

```
SELECT [Customers].[DisplayName], COUNT([Issues].[ID]) AS [CountOfID]
FROM [dbo].[Customers]
LEFT OUTER JOIN [dbo].[Issues] ON [Customers].[ID] = [Issues].[For Customer]
GROUP BY [Customers].[DisplayName]
```

Query AXL:

```
<?xml version="1.0" encoding="utf-8"?>
<Query xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application">

  <References>
    <Reference Source="Customers"></Reference>
    <Reference Source="Issues"></Reference>
  </References>
```

```
  <Results>
    <Property Source="Customers" Name="DisplayName"></Property>
    <Property Alias="CountOfID">
      <Expression>
        <Original>Count(Issues.ID)</Original>
        <FunctionCall Name="Count">
          <Identifier Name="Issues.ID" Index="0"></Identifier>
        </FunctionCall>
      </Expression>
       </Property>
  </Results>

  <Joins>
    <Join Type="Left Outer" Left="Customers" LeftProperty="ID" Right="Issues"
RightProperty="For Customer"></Join>
  </Joins>

  <Groups>
    <Group Source="Customers" Name="DisplayName"></Group>
  </Groups>
</Query>
```

The **References** element (section 2.2.3.7) of the **Query** element (section 2.2.1.1) specifies that the query is using `Customers` and `Issues` as source tables.

The **Results** element (section 2.2.3.9) specifies that the return fields will be `Customers.DisplayName` and the count of `Issues.ID`. The second **Property** element under the **Results** element specifies the expression that returns the count of the number of rows (1) that have a value for `Issues.ID`.

The **Joins** element (section 2.2.3.11) specifies the join relationship. The `Customers` and `Issues` tables are joined by `Customers.ID` and `Issues.[For Customer]`. The **Type** attribute specifies the type of join between the tables. The **Left** attribute specifies the left table of the join. The **Right** attribute specifies the right table of the join.

The **Groups** element (section 2.2.3.74) specifies that the query will group the result set by the `Customers.DisplayName` field.

4 Security

4.1 Security Considerations for Implementers

None.

4.2 Index of Security Fields

None.

5 Appendix A: Full XML Schema

For ease of implementation, this section provides the full XML Schema to which the Access Application Transfer Protocol Structure Version 2 conforms.

5.1 http://schemas.microsoft.com/office/accessservices/2010/12/application Schema

```
<xsd:schema xmlns="http://schemas.microsoft.com/office/accessservices/2010/12/application"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/office/accessservices/2010/12/application"
xmlns:wbld="http://schemas.microsoft.com/office/word/2004/5/build" wbld:hash="true"
wbld:prefix="axl" wbld:cname="AXL" elementFormDefault="qualified">
  <xsd:simpleType name="ST_Decimal">
    <xsd:restriction base="xsd:decimal">
      <xsd:totalDigits value="28"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_ObjectName">
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_UserInterfaceObjectEvent">
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
      <xsd:enumeration value="OnClick" wbld:cname="OnClick"/>
      <xsd:enumeration value="onclick" wbld:cname="onclick"/>
      <xsd:enumeration value="AfterUpdate" wbld:cname="AfterUpdate"/>
      <xsd:enumeration value="afterupdate" wbld:cname="afterupdate"/>
      <xsd:enumeration value="OnCurrent" wbld:cname="OnCurrent"/>
      <xsd:enumeration value="oncurrent" wbld:cname="oncurrent"/>
      <xsd:enumeration value="OnLoad" wbld:cname="OnLoad"/>
      <xsd:enumeration value="onload" wbld:cname="onload"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_DataObjectEvent">
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
      <xsd:enumeration value="AfterInsert" wbld:cname="AfterInsert"/>
      <xsd:enumeration value="AfterUpdate" wbld:cname="AfterUpdate"/>
      <xsd:enumeration value="AfterDelete" wbld:cname="AfterDelete"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_ShortString">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_LongString">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="1024"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="CT_NamedExpression">
```

```
      <xsd:complexContent>
        <xsd:extension base="CT_ExpressionContents">
          <xsd:attribute name="Name" type="xsd:string" wbld:cname="Name" use="required"/>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Expression">
      <xsd:sequence>
        <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExpressionContents">
      <xsd:sequence>
        <xsd:element name="Original" type="xsd:string" minOccurs="0" wbld:cname="Original"/>
        <xsd:choice>
          <xsd:element name="FunctionCall" type="CT_FunctionCall" wbld:cname="FunctionCall"/>
          <xsd:element name="Identifier" type="CT_Identifier" wbld:cname="Identifier"/>
          <xsd:element name="DecimalLiteral" type="CT_DecimalLiteral"
  wbld:cname="DecimalLiteral"/>
          <xsd:element name="IntegerLiteral" type="CT_IntegerLiteral"
  wbld:cname="IntegerLiteral"/>
          <xsd:element name="NullLiteral" type="CT_NullLiteral" wbld:cname="NullLiteral"/>
          <xsd:element name="StringLiteral" type="CT_StringLiteral"
  wbld:cname="StringLiteral"/>
          <xsd:element name="DateTimeLiteral" type="CT_DateTimeLiteral"
  wbld:cname="DateTimeLiteral"/>
          <xsd:element name="DateLiteral" type="CT_DateLiteral" wbld:cname="DateLiteral"/>
          <xsd:element name="TimeLiteral" type="CT_TimeLiteral" wbld:cname="TimeLiteral"/>
          <xsd:element name="BitLiteral" type="CT_BitLiteral" wbld:cname="BitLiteral"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FunctionCall">
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:choice>
          <xsd:element name="FunctionCall" type="CT_IndexedFunctionCall"
  wbld:cname="FunctionCall"/>
          <xsd:element name="Identifier" type="CT_IndexedIdentifier" wbld:cname="Identifier"/>
          <xsd:element name="DecimalLiteral" type="CT_IndexedDecimalLiteral"
  wbld:cname="DecimalLiteral"/>
          <xsd:element name="IntegerLiteral" type="CT_IndexedIntegerLiteral"
  wbld:cname="IntegerLiteral"/>
          <xsd:element name="NullLiteral" type="CT_IndexedNullLiteral"
  wbld:cname="NullLiteral"/>
          <xsd:element name="StringLiteral" type="CT_IndexedStringLiteral"
  wbld:cname="StringLiteral"/>
          <xsd:element name="TypeLiteral" type="CT_IndexedTypeLiteral"
  wbld:cname="TypeLiteral"/>
          <xsd:element name="DatePartLiteral" type="CT_IndexedDatePartLiteral"
  wbld:cname="DatePartLiteral"/>
          <xsd:element name="DateTimeLiteral" type="CT_IndexedDateTimeLiteral"
  wbld:cname="DateTimeLiteral"/>
          <xsd:element name="DateLiteral" type="CT_IndexedDateLiteral"
  wbld:cname="DateLiteral"/>
          <xsd:element name="TimeLiteral" type="CT_IndexedTimeLiteral"
  wbld:cname="TimeLiteral"/>
          <xsd:element name="BitLiteral" type="CT_IndexedBitLiteral" wbld:cname="BitLiteral"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="Name" type="xsd:string" use="required" wbld:cname="Name"/>
```

```
</xsd:complexType>
<xsd:complexType name="CT_Identifier">
  <xsd:attribute name="Name" type="xsd:string" use="required" wbld:cname="Name"/>
</xsd:complexType>
<xsd:complexType name="CT_DecimalLiteral">
  <xsd:attribute name="Value" type="ST_Decimal" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_IntegerLiteral">
  <xsd:attribute name="Value" type="xsd:int" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_NullLiteral"/>
<xsd:complexType name="CT_StringLiteral">
  <xsd:attribute name="Value" type="xsd:string" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_DateTimeLiteral">
  <xsd:attribute name="Value" type="xsd:dateTime" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_TimeLiteral">
  <xsd:attribute name="Value" type="xsd:time" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_DateLiteral">
  <xsd:attribute name="Value" type="xsd:date" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_BitLiteral">
  <xsd:attribute name="Value" type="xsd:boolean" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_IndexedFunctionCall">
  <xsd:complexContent>
    <xsd:extension base="CT_FunctionCall">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedIdentifier">
  <xsd:complexContent>
    <xsd:extension base="CT_Identifier">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedDecimalLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_DecimalLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedIntegerLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_IntegerLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedNullLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_NullLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
```

```
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedStringLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_StringLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="ST_DatePartLiteral">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="YEAR" wbld:cname="YEAR"/>
    <xsd:enumeration value="QUARTER" wbld:cname="QUARTER"/>
    <xsd:enumeration value="MONTH" wbld:cname="MONTH"/>
    <xsd:enumeration value="DAYOFYEAR" wbld:cname="DAYOFYEAR"/>
    <xsd:enumeration value="DAY" wbld:cname="DAY"/>
    <xsd:enumeration value="WEEK" wbld:cname="WEEK"/>
    <xsd:enumeration value="WEEKDAY" wbld:cname="WEEKDAY"/>
    <xsd:enumeration value="HOUR" wbld:cname="HOUR"/>
    <xsd:enumeration value="MINUTE" wbld:cname="MINUTE"/>
    <xsd:enumeration value="SECOND" wbld:cname="SECOND"/>
    <xsd:enumeration value="MILLISECOND" wbld:cname="MILLISECOND"/>
    <xsd:enumeration value="ISO_WEEK" wbld:cname="ISO_WEEK"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CT_IndexedDatePartLiteral">
  <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
  <xsd:attribute name="Value" type="ST_DatePartLiteral" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:simpleType name="ST_TypeLiteral">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="FLOAT" wbld:cname="FLOAT"/>
    <xsd:enumeration value="INTEGER" wbld:cname="INTEGER"/>
    <xsd:enumeration value="CURRENCY" wbld:cname="CURRENCY"/>
    <xsd:enumeration value="YESNO" wbld:cname="YESNO"/>
    <xsd:enumeration value="TEXT" wbld:cname="TEXT"/>
    <xsd:enumeration value="SHORTTEXT" wbld:cname="SHORTTEXT"/>
    <xsd:enumeration value="LONGTEXT" wbld:cname="LONGTEXT"/>
    <xsd:enumeration value="DATEWITHTIME" wbld:cname="DATEWITHTIME"/>
    <xsd:enumeration value="DATE" wbld:cname="DATE"/>
    <xsd:enumeration value="TIME" wbld:cname="TIME"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CT_IndexedTypeLiteral">
  <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
  <xsd:attribute name="Value" type="ST_TypeLiteral" use="required" wbld:cname="Value"/>
</xsd:complexType>
<xsd:complexType name="CT_IndexedDateTimeLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_DateTimeLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedTimeLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_TimeLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
```

```
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedDateLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_DateLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_IndexedBitLiteral">
  <xsd:complexContent>
    <xsd:extension base="CT_BitLiteral">
      <xsd:attribute name="Index" type="xsd:int" use="required" wbld:cname="Index"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="ST_JoinDirection">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Inner" wbld:cname="Inner"/>
    <xsd:enumeration value="Left Outer" wbld:cname="LeftOuter"/>
    <xsd:enumeration value="Right Outer" wbld:cname="RightOuter"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_FieldType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Text" wbld:cname="Text"/>
    <xsd:enumeration value="Number" wbld:cname="Number"/>
    <xsd:enumeration value="Yes/No" wbld:cname="YesNo"/>
    <xsd:enumeration value="Currency" wbld:cname="Currency"/>
    <xsd:enumeration value="Date/Time" wbld:cname="DateTime"/>
    <xsd:enumeration value="LongText" wbld:cname="LongText"/>
    <xsd:enumeration value="Date" wbld:cname="Date"/>
    <xsd:enumeration value="Time" wbld:cname="Time"/>
    <xsd:enumeration value="Integer" wbld:cname="Integer"/>
    <xsd:enumeration value="Decimal" wbld:cname="Decimal"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_SortDirection">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Ascending" wbld:cname="Ascending"/>
    <xsd:enumeration value="Descending" wbld:cname="Descending"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_QuerySourceType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Table" wbld:cname="Table"/>
    <xsd:enumeration value="Query" wbld:cname="Query"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_TextType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SingleLine" wbld:cname="SingleLine"/>
    <xsd:enumeration value="MultipleLines" wbld:cname="MultipleLines"/>
    <xsd:enumeration value="Hyperlink" wbld:cname="Hyperlink"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ST_DecimalPlaces">
  <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:minInclusive value="0"/>
```

```
      <xsd:maxInclusive value="15"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:attribute name="BoundColumn" wbld:cname="BoundColumn" type="ST_ObjectName"/>
  <xsd:attribute name="Caption" type="ST_LongString" wbld:cname="Caption"/>
  <xsd:attribute name="CurrencyLocale" wbld:cname="CurrencyLocale" type="xsd:string"/>
  <xsd:attribute name="CurrencySymbol" wbld:cname="CurrencySymbol" type="xsd:string"/>
  <xsd:attribute name="DecimalPlaces" type="ST_DecimalPlaces" wbld:cname="DecimalPlaces"/>
  <xsd:attribute name="Description" wbld:cname="Description" type="xsd:string"/>
  <xsd:attribute name="Direction" wbld:cname="Direction" type="ST_SortDirection"/>
  <xsd:attribute name="DisplayColumn" wbld:cname="DisplayColumn" type="ST_ObjectName"/>
  <xsd:attribute name="DisplayName" wbld:cname="DisplayName" type="ST_ObjectName"/>
  <xsd:attribute name="DisplayOrder" wbld:cname="DisplayOrder" type="xsd:unsignedLong"/>
  <xsd:attribute name="Format" wbld:cname="Format" type="ST_Format"/>
  <xsd:attribute name="InternalName" wbld:cname="InternalName" type="ST_ObjectName"/>
  <xsd:attribute name="LookupType" wbld:cname="LookupType" type="ST_LookupType"/>
  <xsd:attribute name="ObjectId" wbld:cname="ObjectId" type="ST_ObjectName"/>
  <xsd:attribute name="ReferencedTable" wbld:cname="ReferencedTable" type="ST_ObjectName"/>
  <xsd:attribute name="SiteURL" wbld:cname="SiteURL" type="xsd:anyURI"/>
  <xsd:attribute name="StoreGeneratedPattern" wbld:cname="StoreGeneratedPattern"
type="TStoreGeneratedPattern"/>
  <xsd:attribute name="TextType" wbld:cname="TextType" type="ST_TextType"/>
  <xsd:attribute name="UnderlyingType" wbld:cname="UnderlyingType" type="ST_FieldType"/>
  <xsd:attribute name="Values" wbld:cname="Values" type="xsd:string"/>
  <xsd:simpleType name="TStoreGeneratedPattern" wbld:cname="TStoreGeneratedPattern">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="None" wbld:cname="None"/>
      <xsd:enumeration value="Identity" wbld:cname="Identity"/>
      <xsd:enumeration value="Computed" wbld:cname="Computed"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ST_LookupType" wbld:cname="ST_LookupType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TableOrQuery" wbld:cname="TableOrQuery"/>
      <xsd:enumeration value="ValueList" wbld:cname="ValueList"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="CT_Index">
    <xsd:sequence>
      <xsd:element name="PropertyRef" type="CT_PropertyRefIndex" wbld:cname="PropertyRef"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"
form="qualified"/>
    <xsd:attribute name="ObjectId" wbld:cname="ObjectId" type="ST_ObjectName"
form="qualified"/>
    <xsd:attribute name="Caption" type="ST_LongString" wbld:cname="Caption"
form="qualified"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Unique">
    <xsd:complexContent>
      <xsd:extension base="CT_Index"/>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="CT_ConstraintWithExpression">
    <xsd:sequence>
      <xsd:element name="PropertyRef" wbld:cname="PropertyRef" type="CT_PropertyRef"
minOccurs="0"/>
      <xsd:element name="Expression" wbld:cname="Expression" type="CT_ExpressionContents"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"
form="qualified"/>
    <xsd:attribute name="ObjectId" wbld:cname="ObjectId" type="ST_ObjectName"
form="qualified"/>
  </xsd:complexType>
  <xsd:complexType name="CT_CheckConstraint">
    <xsd:complexContent>
      <xsd:extension base="CT_ConstraintWithExpression">
        <xsd:attribute name="CheckData" wbld:cname="CheckData" type="xsd:boolean"
use="optional" form="qualified"/>
        <xsd:attribute name="Message" wbld:cname="Message" type="ST_ShortString"
use="optional" form="qualified"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="CT_DefaultConstraint">
    <xsd:complexContent>
      <xsd:restriction base="CT_ConstraintWithExpression">
        <xsd:sequence>
          <xsd:element name="PropertyRef" wbld:cname="PropertyRef" type="CT_PropertyRef"/>
          <xsd:element name="Expression" wbld:cname="Expression"
type="CT_ExpressionContents"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="CT_PropertyRef">
    <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="CT_PropertyRefIndex">
    <xsd:complexContent>
      <xsd:extension base="CT_PropertyRef">
        <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="CT_ExtendedAttribute" wbld:cname="CT_ExtendedAttribute">
    <xsd:attribute name="Name" wbld:cname="Name" type="ST_ObjectName" use="required"/>
    <xsd:attribute name="Value" wbld:cname="Value" type="ST_LongString" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="CT_ExtendedAttributes" wbld:cname="CT_ExtendedAttributes">
    <xsd:sequence>
      <xsd:element name="ExtendedAttribute" wbld:cname="ExtendedAttribute"
type="CT_ExtendedAttribute" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="ExtendedAttributes" wbld:cname="ExtendedAttributes"
type="CT_ExtendedAttributes"/>
  <xsd:simpleType name="ST_Format" wbld:cname="ST_Format">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="General Number" wbld:cname="GeneralNumber"/>
      <xsd:enumeration value="Currency" wbld:cname="Currency"/>
      <xsd:enumeration value="Fixed" wbld:cname="Fixed"/>
      <xsd:enumeration value="Standard" wbld:cname="Standard"/>
      <xsd:enumeration value="Percent" wbld:cname="Percent"/>
      <xsd:enumeration value="General Date" wbld:cname="GeneralDate"/>
      <xsd:enumeration value="Long Date" wbld:cname="LongDate"/>
      <xsd:enumeration value="Short Date" wbld:cname="ShortDate"/>
```

```
      <xsd:enumeration value="Long Time" wbld:cname="LongTime"/>
      <xsd:enumeration value="Short Time" wbld:cname="ShortTime"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="Index" wbld:cname="Index" type="CT_Index"/>
  <xsd:element name="Unique" wbld:cname="Unique" type="CT_Unique"/>
  <xsd:element name="CheckConstraint" wbld:cname="CheckConstraint"
type="CT_CheckConstraint"/>
  <xsd:element name="DefaultConstraint" wbld:cname="DefaultConstraint"
type="CT_DefaultConstraint"/>
  <xsd:element name="PropertyRef" wbld:cname="PropertyRef" type="CT_PropertyRef"/>
  <xsd:complexType name="CT_Order">
    <xsd:attribute name="Name" type="ST_LongString" use="required" wbld:cname="Name"/>
    <xsd:attribute name="Source" type="ST_ObjectName" use="required" wbld:cname="Source"/>
    <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_OrderExpression">
    <xsd:all>
      <xsd:element name="Expression" type="CT_ExpressionContents" minOccurs="1"
wbld:cname="Expression"/>
    </xsd:all>
    <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Ordering">
    <xsd:choice maxOccurs="255">
      <xsd:element name="Order" wbld:cname="Order" type="CT_Order"/>
      <xsd:element name="OrderExpression" wbld:cname="OrderExpression"
type="CT_OrderExpression"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_AdHocOrder">
    <xsd:attribute name="Name" type="ST_LongString" use="required" wbld:cname="Name"/>
    <xsd:attribute name="Direction" type="ST_SortDirection" default="Ascending"
wbld:cname="Direction" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_AdHocOrdering">
    <xsd:choice maxOccurs="255">
      <xsd:element name="Order" wbld:cname="Order" type="CT_AdHocOrder"/>
      <xsd:element name="OrderExpression" wbld:cname="OrderExpression"
type="CT_OrderExpression"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_Reference">
    <xsd:all>
      <xsd:element name="ReferenceParameters" minOccurs="0" type="CT_Parameters"
wbld:cname="ReferenceParameters"/>
    </xsd:all>
    <xsd:attribute name="Source" type="ST_ObjectName" use="required" wbld:cname="Source"/>
    <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
    <xsd:attribute name="Type" type="ST_QuerySourceType" default="Table" wbld:cname="Type"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_References">
    <xsd:sequence>
      <xsd:element name="Reference" maxOccurs="unbounded" wbld:cname="Reference"
type="CT_Reference"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
  <xsd:complexType name="CT_Result">
    <xsd:all>
      <xsd:element name="Expression" type="CT_ExpressionContents" minOccurs="0"
wbld:cname="Expression"/>
    </xsd:all>
    <xsd:attribute name="Source" type="ST_ObjectName" wbld:cname="Source" use="optional"/>
    <xsd:attribute name="Name" type="ST_LongString" wbld:cname="Name" use="optional"/>
    <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
    <xsd:attribute name="All" type="xsd:boolean" default="false" wbld:cname="All"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Results">
    <xsd:sequence>
      <xsd:element name="Property" maxOccurs="255" wbld:cname="Property" type="CT_Result"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_Join">
    <xsd:attribute name="Left" type="ST_ObjectName" use="required" wbld:cname="Left"/>
    <xsd:attribute name="Right" type="ST_ObjectName" use="required" wbld:cname="Right"/>
    <xsd:attribute name="LeftProperty" type="ST_LongString" use="required"
wbld:cname="LeftProperty"/>
    <xsd:attribute name="RightProperty" type="ST_LongString" use="required"
wbld:cname="RightProperty"/>
    <xsd:attribute name="Type" type="ST_JoinDirection" default="Inner" wbld:cname="Type"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Joins">
    <xsd:sequence>
      <xsd:element name="Join" wbld:cname="Join" maxOccurs="unbounded" type="CT_Join"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_Group">
    <xsd:attribute name="Source" type="ST_ObjectName" wbld:cname="Source" use="required"/>
    <xsd:attribute name="Name" type="ST_LongString" wbld:cname="Name" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="CT_GroupExpression">
    <xsd:all>
      <xsd:element name="Expression" type="CT_ExpressionContents" minOccurs="1"
wbld:cname="Expression"/>
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="CT_Groups">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="Group" wbld:cname="Group" type="CT_Group"/>
      <xsd:element name="GroupExpression" wbld:cname="GroupExpression"
type="CT_GroupExpression"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_Parameter">
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
    <xsd:attribute name="Type" type="ST_FieldType" use="required" wbld:cname="Type"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Parameters">
    <xsd:sequence>
      <xsd:element name="Parameter" maxOccurs="unbounded" wbld:cname="Parameter"
type="CT_Parameter"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_ParameterDefinition">
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
```

```
    <xsd:attribute name="Description" type="ST_LongString" wbld:cname="Description"
use="optional"/>
    <xsd:attribute name="Type" type="ST_FieldType" use="required" wbld:cname="Type"/>
  </xsd:complexType>
  <xsd:complexType name="CT_ParameterDefinitions">
    <xsd:sequence>
      <xsd:element name="Parameter" maxOccurs="unbounded" wbld:cname="Parameter"
type="CT_ParameterDefinition"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_ParameterValue">
    <xsd:sequence>
      <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
    </xsd:sequence>
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
  </xsd:complexType>
  <xsd:complexType name="CT_OutputParameter">
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
    <xsd:attribute name="LocalVarName" type="ST_ObjectName" use="required"
wbld:cname="LocalVarName"/>
  </xsd:complexType>
  <xsd:complexType name="CT_ParameterValues">
    <xsd:sequence>
      <xsd:element name="Parameter" minOccurs="0" maxOccurs="unbounded"
wbld:cname="Parameter" type="CT_ParameterValue"/>
      <xsd:element name="OutputParameter" minOccurs="0" maxOccurs="unbounded"
wbld:cname="OutputParameter" type="CT_OutputParameter"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_TopRows">
    <xsd:attribute name="Rows" type="xsd:positiveInteger" use="required" wbld:cname="Rows"/>
  </xsd:complexType>
  <xsd:simpleType name="ST_Percent">
    <xsd:restriction base="xsd:float">
      <xsd:minExclusive value="0"/>
      <xsd:maxInclusive value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="CT_TopPercent">
    <xsd:attribute name="Percent" use="required" wbld:cname="Percent" type="ST_Percent"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Query">
    <xsd:sequence>
      <xsd:choice minOccurs="0">
        <xsd:element name="TopRows" type="CT_TopRows" wbld:cname="TopRows"/>
        <xsd:element name="TopPercent" type="CT_TopPercent" wbld:cname="TopPercent"/>
      </xsd:choice>
      <xsd:element name="Parameters" minOccurs="0" type="CT_Parameters"
wbld:cname="Parameters"/>
      <xsd:element name="References" type="CT_References" wbld:cname="References"/>
      <xsd:element name="Results" type="CT_Results" wbld:cname="Results"/>
      <xsd:element name="Joins" minOccurs="0" type="CT_Joins" wbld:cname="Joins"/>
      <xsd:element name="Restriction" type="CT_Expression" minOccurs="0"
wbld:cname="Restriction"/>
      <xsd:element name="Groups" minOccurs="0" type="CT_Groups" wbld:cname="Groups"/>
      <xsd:element name="GroupRestriction" minOccurs="0" type="CT_Expression"
wbld:cname="GroupRestriction"/>
      <xsd:element name="Ordering" minOccurs="0" type="CT_Ordering" wbld:cname="Ordering"/>
    </xsd:sequence>
    <xsd:attribute name="Name" type="ST_ObjectName" wbld:cname="Name" use="optional"/>
```

```
    <xsd:attribute name="Distinct" type="xsd:boolean" wbld:cname="Distinct" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Argument">
    <xsd:simpleContent>
      <xsd:extension base="ST_LongString">
        <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="CT_ExpressionArgument">
    <xsd:sequence>
      <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
    </xsd:sequence>
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
  </xsd:complexType>
  <xsd:complexType name="CT_Action">
    <xsd:sequence>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Argument" wbld:cname="Argument" type="CT_Argument"/>
        <xsd:element name="ExpressionArgument" wbld:cname="ExpressionArgument"
type="CT_ExpressionArgument"/>
      </xsd:choice>
      <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
    </xsd:sequence>
    <xsd:attribute name="Name" type="ST_ObjectName" use="required" wbld:cname="Name"/>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_ForEachRecordData">
    <xsd:sequence>
      <xsd:element name="Reference" type="ST_ObjectName" wbld:cname="Reference"/>
      <xsd:element name="WhereCondition" minOccurs="0" type="CT_Expression"
wbld:cname="WhereCondition"/>
      <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
    </xsd:sequence>
    <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_ForEachRecord">
    <xsd:sequence>
      <xsd:element name="Data" wbld:cname="Data" type="CT_ForEachRecordData"/>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_LookupRecordData">
    <xsd:sequence>
      <xsd:element name="Reference" type="ST_ObjectName" wbld:cname="Reference"/>
      <xsd:element name="WhereCondition" minOccurs="0" type="CT_Expression"
wbld:cname="WhereCondition"/>
      <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
    </xsd:sequence>
    <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_LookupRecord">
    <xsd:sequence>
      <xsd:element name="Data" wbld:cname="Data" type="CT_LookupRecordData"/>
```

```
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_CreateRecordData">
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="Reference" type="ST_ObjectName" wbld:cname="Reference"/>
      </xsd:choice>
      <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterValues"
wbld:cname="Parameters"/>
    </xsd:sequence>
    <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_CreateRecord">
    <xsd:sequence>
      <xsd:element name="Data" wbld:cname="Data" type="CT_CreateRecordData"/>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_EditRecordData">
    <xsd:attribute name="Alias" type="ST_ObjectName" wbld:cname="Alias" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_EditRecord">
    <xsd:sequence>
      <xsd:element name="Data" type="CT_EditRecordData" wbld:cname="Data"/>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_UserInterfaceIf">
    <xsd:sequence>
      <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
      <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_UserInterfaceElseIf">
    <xsd:sequence>
      <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
      <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_UserInterfaceElse">
    <xsd:sequence>
      <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
```

```
  <xsd:complexType name="CT_DataIf">
    <xsd:sequence>
      <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_DataElseIf">
    <xsd:sequence>
      <xsd:element name="Condition" type="CT_Expression" wbld:cname="Condition"/>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_DataElse">
    <xsd:sequence>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_UserInterfaceConditionalBlock">
    <xsd:sequence>
      <xsd:element name="If" wbld:cname="If" type="CT_UserInterfaceIf"/>
      <xsd:element name="ElseIf" minOccurs="0" maxOccurs="unbounded" wbld:cname="ElseIf"
type="CT_UserInterfaceElseIf"/>
      <xsd:element name="Else" minOccurs="0" wbld:cname="Else" type="CT_UserInterfaceElse"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_DataConditionalBlock">
    <xsd:sequence>
      <xsd:element name="If" wbld:cname="If" type="CT_DataIf"/>
      <xsd:element name="ElseIf" minOccurs="0" maxOccurs="unbounded" wbld:cname="ElseIf"
type="CT_DataElseIf"/>
      <xsd:element name="Else" minOccurs="0" wbld:cname="Else" type="CT_DataElse"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CT_UserInterfaceMacroStatements">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Action" type="CT_Action" wbld:cname="Action"/>
      <xsd:element name="Comment" type="ST_LongString" wbld:cname="Comment"/>
      <xsd:element name="ConditionalBlock" type="CT_UserInterfaceConditionalBlock"
wbld:cname="ConditionalBlock"/>
      <xsd:element name="StatementGroup" type="CT_UserInterfaceMacroStatementGroup"
wbld:cname="StatementGroup"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_DataMacroStatements">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Action" type="CT_Action" wbld:cname="Action"/>
      <xsd:element name="Comment" type="ST_LongString" wbld:cname="Comment"/>
      <xsd:element name="ForEachRecord" type="CT_ForEachRecord" wbld:cname="ForEachRecord"/>
      <xsd:element name="LookUpRecord" type="CT_LookupRecord" wbld:cname="LookupRecord"/>
      <xsd:element name="CreateRecord" type="CT_CreateRecord" wbld:cname="CreateRecord"/>
      <xsd:element name="EditRecord" type="CT_EditRecord" wbld:cname="EditRecord"/>
      <xsd:element name="ConditionalBlock" type="CT_DataConditionalBlock"
wbld:cname="ConditionalBlock"/>
```

```
      <xsd:element name="StatementGroup" type="CT_DataMacroStatementGroup"
wbld:cname="StatementGroup"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_UserInterfaceMacroStatementGroup">
    <xsd:sequence>
      <xsd:element name="Statements" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Description" type="ST_LongString" wbld:cname="Description"
use="optional"/>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_DataMacroStatementGroup">
    <xsd:sequence>
      <xsd:element name="Statements" type="CT_DataMacroStatements" wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Description" type="ST_LongString" wbld:cname="Description"
use="optional"/>
    <xsd:attribute name="Collapsed" type="xsd:boolean" wbld:cname="Collapsed"
use="optional"/>
  </xsd:complexType>
  <xsd:element name="Query" type="CT_Query" wbld:cname="Query"/>
  <xsd:element name="Ordering" type="CT_AdHocOrdering" wbld:cname="Ordering"/>
  <xsd:complexType name="CT_Expressions">
    <xsd:sequence>
      <xsd:element name="Expression" minOccurs="0" maxOccurs="unbounded"
type="CT_NamedExpression" wbld:cname="Expression"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="Expression" type="CT_ExpressionContents" wbld:cname="Expression"/>
  <xsd:complexType name="CT_UserInterfaceMacros">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="UserInterfaceMacro" minOccurs="0" maxOccurs="unbounded"
type="CT_UserInterfaceMacro" wbld:cname="UserInterfaceMacro"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_DataMacros">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="DataMacro" minOccurs="0" maxOccurs="unbounded" type="CT_DataMacro"
wbld:cname="DataMacro"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:complexType name="CT_EventDataMacro">
    <xsd:sequence>
      <xsd:element name="DataMacro" minOccurs="1" maxOccurs="1" type="CT_DataMacro"
wbld:cname="DataMacro"/>
    </xsd:sequence>
    <xsd:attribute name="Version" type="xsd:string" wbld:cname="Version" use="optional"/>
  </xsd:complexType>
  <xsd:element name="UserInterfaceMacros" wbld:cname="UserInterfaceMacros"
type="CT_UserInterfaceMacros"/>
  <xsd:element name="DataMacros" wbld:cname="DataMacros" type="CT_DataMacros"/>
  <xsd:element name="EventDataMacro" wbld:cname="EventDataMacro" type="CT_EventDataMacro"/>
  <xsd:element name="UserInterfaceMacro" wbld:cname="UserInterfaceMacro"
type="CT_UserInterfaceMacro"/>
  <xsd:element name="DataMacro" wbld:cname="DataMacro" type="CT_DataMacro"/>
  <xsd:complexType name="CT_UserInterfaceMacro">
    <xsd:sequence>
```

```
      <xsd:element name="Statements" minOccurs="0" type="CT_UserInterfaceMacroStatements"
wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="For" type="ST_ObjectName" wbld:cname="For" use="optional"/>
    <xsd:attribute name="Event" type="ST_UserInterfaceObjectEvent" wbld:cname="Event"
use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_DataMacro">
    <xsd:sequence>
      <xsd:element name="Parameters" minOccurs="0" type="CT_ParameterDefinitions"
wbld:cname="Parameters"/>
      <xsd:element name="Statements" minOccurs="0" type="CT_DataMacroStatements"
wbld:cname="Statements"/>
    </xsd:sequence>
    <xsd:attribute name="Event" type="ST_DataObjectEvent" wbld:cname="Event" use="optional"/>
    <xsd:attribute name="Version" type="xsd:string" wbld:cname="Version" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="CT_FormDef">
    <xsd:sequence>
      <xsd:element name="RecordSource" type="xsd:string" wbld:cname="RecordSource"
minOccurs="0"/>
      <xsd:element name="Html" wbld:cname="Html" type="xsd:string"/>
      <xsd:element name="UserInterfaceMacros" wbld:cname="UserInterfaceMacros"
wbld:codeName="UserInterfaceMacros" type="CT_UserInterfaceMacros" minOccurs="0"/>
      <xsd:element name="Expressions" wbld:cname="Expressions" wbld:codeName="Expressions"
type="CT_Expressions" minOccurs="0"/>
      <xsd:element name="Query" wbld:cname="Query" wbld:codeName="Query" type="CT_Query"
minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="FormDef" wbld:cname="FormDef" type="CT_FormDef"/>
</xsd:schema>
```

6 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Access 2013
- Microsoft SharePoint Server 2013

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.1.4: Access Services preserves these values in SharePoint Server 2013.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

A

After insert example 110
Applicability 12
Attributes 22

C

Change tracking 133
Common data types and fields 13
Complex type
  CT_UserInterfaceMcroStatements 50
Complex types 27
  CT_Action 39
  CT_AdHocOrder 29
  CT_AdHocOrdering 29
  CT_Argument (section 2.2.3.19 38, section 2.2.3.78 78)
  CT_CreateRecord 43
  CT_CreateRecordData 42
  CT_DataConditionalBlock 49
  CT_DataElse 48
  CT_DataElseIf 47
  CT_DataIf 47
  CT_DataMacro 54
  CT_DataMacros 53
  CT_DataMacroStatementGroup 52
  CT_DataMacroStatements 51
  CT_EditRecord 44
  CT_EditRecordData 44
  CT_ForEachRecord 40
  CT_ForEachRecordData 40
  CT_Join 33
  CT_Joins 34
  CT_LookupRecord 42
  CT_LookupRecordData 41
  CT_Order (section 2.2.3.2 27, section 2.2.3.72 75, section 2.2.3.73 76, section 2.2.3.76 78)
  CT_Ordering (section 2.2.3.3 28, section 2.2.3.5 29, section 2.2.3.74 76)
  CT_Parameter (section 2.2.3.12 34, section 2.2.3.72 75, section 2.2.3.73 76)
  CT_ParameterDefinitions (section 2.2.3.14 35, section 2.2.3.15 35)
  CT_Parameters (section 2.2.3.13 34, section 2.2.3.79 79)
  CT_ParameterValue (section 2.2.3.16 36, section 2.2.3.75 77)
  CT_ParameterValues 36
  CT_Query 37
  CT_Reference 30
  CT_References (section 2.2.3.7 31, section 2.2.3.7 31)
  CT_Result 31
  CT_Results 32
  CT_UserInterfaceConditionalBlock 48
  CT_UserInterfaceElse 46
  CT_UserInterfaceElseIf 45
  CT_UserInterfaceIf 45
  CT_UserInterfaceMacro 54
  CT_UserInterfaceMacros 53
  CT_UserInterfaceMacroStatementGroup 52
Conceptual overview
  ignored value 17
  macros 15
  queries 16
  tables 13
CT_Action complex type 39
CT_AdHocOrder complex type 29
CT_AdHocOrdering complex type 29
CT_Argument complex type (section 2.2.3.19 38, section 2.2.3.78 78)
CT_CreateRecord complex types 43
CT_CreateRecordData complex type 42
CT_DataConditionalBlock complex type 49
CT_DataElse complex type 48
CT_DataElseIf complex type 47
CT_DataIf complex type 47
CT_DataMacro complex type 54
CT_DataMacros complex types 53
CT_DataMacroStatementGroup complex type 52
CT_DataMacroStatements complex type 51
CT_EditRecord complex type 44
CT_EditRecordData complex type 44
CT_ForEachRecord complex type 40
CT_ForEachRecordData complex type 40
CT_Join complex type 33
CT_Joins complex type 34
CT_LookupRecord complex type 42
CT_LookupRecordData complex type 41
CT_Order complex type (section 2.2.3.2 27, section 2.2.3.72 75, section 2.2.3.73 76, section 2.2.3.76 78)
CT_Ordering complex type (section 2.2.3.3 28, section 2.2.3.5 29, section 2.2.3.74 76)
CT_Parameter complex type (section 2.2.3.12 34, section 2.2.3.72 75, section 2.2.3.73 76)
CT_ParameterDefinitions complex type (section 2.2.3.14 35, section 2.2.3.15 35)
CT_Parameters complex type (section 2.2.3.13 34, section 2.2.3.79 79)
CT_ParameterValue complex type (section 2.2.3.16 36, section 2.2.3.75 77)
CT_ParameterValues complex type 36
CT_Query complex type 37
CT_Reference complex type 30
CT_References complex type (section 2.2.3.7 31, section 2.2.3.7 31)
CT_Result complex type 31
CT_Results complex type 32
CT_UserInterfaceConditionalBlock complex type 48
CT_UserInterfaceElse complex type 46
CT_UserInterfaceElseIf complex type 45
CT_UserInterfaceIf complex type 45
CT_UserInterfaceMacro complex types 54
CT_UserInterfaceMacros complex type 53
CT_UserInterfaceMacroStatementGroup complex type 52

CT_UserInterfaceMcroStatements complex type 50

D

Data macros
overview 11
Data Macros example 109
Data types and fields - common 13
DataMacros element 18
Details
attributes 22
common data types and fields 13
complex types 27
CT_Action complex type 39
CT_AdHocOrder complex type 29
CT_AdHocOrdering complex type 29
CT_Argument complex type (section 2.2.3.19 38, section 2.2.3.78 78)
CT_CreateRecord complex types 43
CT_CreateRecordData complex type 42
CT_DataConditionalBlock complex type 49
CT_DataElse complex type 48
CT_DataElseIf complex type 47
CT_DataIf complex type 47
CT_DataMacro complex type 54
CT_DataMacros complex types 53
CT_DataMacroStatementGroup complex type 52
CT_DataMacroStatements complex type 51
CT_EditRecord complex type 44
CT_EditRecordData complex type 44
CT_ForEachRecord complex type 40
CT_ForEachRecordData complex type 40
CT_Join complex type 33
CT_Joins complex type 34
CT_LookupRecord complex type 42
CT_LookupRecordData complex type 41
CT_Order complex type (section 2.2.3.2 27, section 2.2.3.72 75, section 2.2.3.73 76, section 2.2.3.76 78)
CT_Ordering complex type (section 2.2.3.3 28, section 2.2.3.5 29, section 2.2.3.74 76)
CT_Parameter complex type (section 2.2.3.12 34, section 2.2.3.72 75, section 2.2.3.73 76)
CT_ParameterDefinitions complex type (section 2.2.3.14 35, section 2.2.3.15 35)
CT_Parameters complex type (section 2.2.3.13 34, section 2.2.3.79 79)
CT_ParameterValue complex type (section 2.2.3.16 36, section 2.2.3.75 77)
CT_ParameterValues complex type 36
CT_Query complex type 37
CT_Reference complex type 30
CT_References complex type (section 2.2.3.7 31, section 2.2.3.7 31)
CT_Result complex type 31
CT_Results complex type 32
CT_UserInterfaceConditionalBlock complex type 48
CT_UserInterfaceElse complex type 46
CT_UserInterfaceElseIf complex type 45
CT_UserInterfaceIf complex type 45
CT_UserInterfaceMacro complex types 54
CT_UserInterfaceMacros complex type 53
CT_UserInterfaceMacroStatementGroup complex type 52
CT_UserInterfaceMcroStatements complex type 50
DataMacros element 18
elements 18
macros 96
Ordering element 18
query element 18
simple types 85
ST_FieldType simple type 87
ST_JoinDirection simple type (section 2.2.4.5 87, section 2.2.4.15 93, section 2.2.4.16 94)
ST_LongString simple type 86
ST_ObjectName simple type 85
ST_ShortString simple type 86
ST_SortDirection simple type (section 2.2.4.7 88, section 2.2.4.18 95)
ST_UserInterfaceObjectEvent simple type 85
UserInterfaceMacro elements 19
UserInterfaceMacros elements 18

E

Elements 18
DataMacros 18
Ordering 18
query 18
UserInterfaceMacro 19
UserInterfaceMacros 18
Examples 105
after insert 110
Data Macros 109
after insert 110
named macro 111
validate delete 109
Forms 107
multiple reference query with join 114
named macro 111
Queries 113
multiple reference query with join 114
single reference query 113
single reference query 113
standalone 109
Tables 105
UI Macros 108
standalone 109
validation and exceptions 108
validate delete 109
validation and exceptions 108

F

Fields - security index 116
Fields - vendor-extensible 12
Forms example 107
Full XML schema 117

G

Glossary 8

I

Ignored value conceptual overview 17
Implementer - security considerations 116
Index of security fields 116
Informative references 9
Introduction 8

L

Localization 12

M

Macros 96
Multiple reference query with join example 114

N

Named macro example 111
Normative references 9

O

Ordering element 18
Overview (synopsis) 9
  data macros 11
  queries 11
  tables 10
  UI macros 10

P

Product behavior 132

Q

Queries
  overview 11
Queries conceptual overview 16
Queries example 113
Query element 18

R

References 9
  informative 9
  normative 9
Relationship to protocols and other structures 12

S

Security
  field index 116
  implementer considerations 116
Simple types 85
  ST_FieldType 87
  ST_JoinDirection (section 2.2.4.5 87, section 2.2.4.15 93, section 2.2.4.16 94)
  ST_LongString 86
  ST_ObjectName 85
  ST_ShortString 86
  ST_SortDirection (section 2.2.4.7 88, section 2.2.4.18 95)
  ST_UserInterfaceObjectEvent 85
Single reference query example 113
ST_FieldType simple type 87
ST_JoinDirection simple type (section 2.2.4.5 87, section 2.2.4.15 93, section 2.2.4.16 94)
ST_LongString simple type 86
ST_ObjectName simple type 85
ST_ShortString simple type 86
ST_SortDirection simple type (section 2.2.4.7 88, section 2.2.4.18 95)
ST_UserInterfaceObjectEvent simple type 85
Standalone example 109
Structures
  arguments 101
  attributes 22
  complex types 27
  data macros 16
  elements 18
  ignored value 17
  macros (section 2.1.2 15, section 2.1.2 15, section 2.2.5 96)
  overview 13
  queries 16
  query 18
  simple types 85
  tables 13
  user interface macros 16
  variables 16

T

Tables
  overview 10
Tables conceptual overview (section 2.1.1 13, section 2.1.2 15)
Tables example 105
Tracking changes 133

U

UI macros
  overview 10
UI Macros example 108
UserInterfaceMacro elements 19
UserInterfaceMacros elements 18

V

Validate delete example 109
Validation and exceptions example 108
Vendor-extensible fields 12
Versioning 12

X

XML schema 117