[MS-DPAD]:
Alert Definition Data Portability Overview

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
07/07/2011	0.1	New	Released new document.
11/03/2011	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/19/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/23/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/27/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
05/24/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
06/29/2012	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.
10/08/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/26/2013	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
06/11/2013	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
12/05/2013	0.1	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

The Alert Definition Data Portability Overview document specifies the components and methodologies that are used for data portability within the Microsoft SQL Server Reporting Services Data Alerts system. This document provides an example user scenario for data export to provide an overview of data portability between a Microsoft SQL Server 2012 Reporting Services Data Alerts feature and a third-party application.

SQL Server 2012 Reporting Services Data Alerts uses a Microsoft SQL Server user database that contains the data for the application. This document provides a data portability scenario for exporting data from SQL Server 2012 Reporting Services Data Alerts. In this scenario, the user must use an API or tools that can consume the data within the third-party application.

In the export data scenario, a user can use a Transact-SQL query to extract the data from the SQL Server 2012 Reporting Services Data Alerts database and save the result. This methodology for exporting data is described in [MS-DPBCP] section 2.1. Data is exported as a set of rows, one row for each Alert Definition. The user can use the output to import the data into a third-party application.

1.1 Glossary

The following terms are specific to this document:

Alert Definition: The set of metadata that enables Reporting Services to process data and send alert messages to users, when desired by the user.

Alerting Database: A database that stores the Alerting Metadata.

1.2 References

[MS-DPBCP] Microsoft Corporation, "Bulk Copy Utility Data Portability Overview".

[MSDN-NLS] See [MSDN-NLSAPIRef].

[MSDN-NLSAPIRef] Microsoft Corporation, "National Language Support (NLS) API Reference", http://msdn.microsoft.com/en-us/goglobal/bb896001.aspx

[XML1.0/5] W3C, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", http://www.w3.org/TR/2008/REC-xml-20081126/

2 Data Portability Scenarios

The data portability scenario described in the following sections describes exporting data by using a Transact-SQL query in Microsoft SQL Server. The scenario describes the export of data from a Microsoft SQL Server 2012 Reporting Services Data Alerts database.

2.1 Third-Party Alerting System Consuming Alert Definitions from the Alerting Database

This scenario describes exporting the Alert Definition metadata from a Microsoft SQL Server 2012 Reporting Services Data Alerts database. The information is retrieved by using a Transact-SQL query, as specified in this document. The Transact-SQL query can be executed from SQL Server Management Studio. To export the data, the user can save the result set from SQL Server Management Studio in any format, such as text or comma separated values (CSV), that is supported by the SQL Server Management Studio tool.

2.1.1 Data Description

An Alert Definition is a set of metadata that defines an alert and consists of the following information:

- Which report data to process.
- Which data fields to evaluate rules against.
- The rules that define business logic.
- Scheduling options.
- Delivery settings.

An Alert Definition is used to process data and send alert messages, when alert messages are required. The Alert Definition data is stored in the Alerting Database, and is created by a Reporting Services alert design tool.

2.1.2 Format and Protocol Summary

No formats or protocols are used in this scenario.

2.1.3 Data Portability Methodology

In this scenario, the Alert Definition data is stored in the Alerting Database as a set of metadata in table columns. To interpret the Alert Definition data that is retrieved from the Alerting Database, see section 2.1.3.5, which describes the Alert Definition metadata.

To extract the data:

1. Connect to the Alerting Database by using a tool that is capable of executing Transact-SQL statements.
2. Use the following Transact-SQL query to obtain the metadata for all Alert Definitions (each row represents a unique Alert Definition).

```
select
```

```
--Alert Definition Name
ad.Name as AlertDefinitionName,
--User Info
u.UserLogonName,
--Report URI
feed.SourceUrl as ReportURI,
--Data Feed
feed.DataFeedUrl as DataFeedURL,
--Data Condition
ad.RuleDefinition,
ad.DefinitionCulture,
ad.RuleDefinition,
--Schedule
sch.StartDate as Sch_StartDate, sch.EndDate as Sch_EndDate, sch.RecurrenceType as
Sch_RecurrenceType, sch.RecurrenceInterval as Sch_RecurrenceInterval,
ad.Attributes,

--Delivery Settings
dt.[Address] as EmailRecipients,
dt.MessageSubject as EmailSubject,
ad.[Description] as EmailDescription

from AlertDefinition ad
join [User] u on ad.OwnerId = u.UserId
join [AlertSchedule] sch on sch.ScheduleId = ad.ScheduleId
join [Feed] feed on feed.FeedId = ad.FeedId
join [DeliveryTarget] dt on dt.AlertDefinitionId = ad.AlertDefinitionID
```

To interpret the Alert Definition data that is retrieved from the Alerting Database, see section 2.1.3.5.1, which describes the Alert Definition metadata.

2.1.3.1 Preconditions

Ensure that the Microsoft SQL Server process that hosts the Alerting Database file is started on the server. Grant the appropriate permissions to the user who is accessing the Alerting Database to read data from the tables within the database.

2.1.3.2 Versioning

None.

2.1.3.3 Error Handling

None.

2.1.3.4 Coherency Requirements

If the system is actively processing requests, data must be exported from all related tables in a single transaction to guarantee referential integrity.

2.1.3.5 Additional Considerations

This section defines the Transact-SQL query that retrieves the Alert Definition metadata.

Column Name	Data Type	Description
AlertDefinitionName	NVarChar(Max)	The name of the Alert Definition, as defined by the user who created the Alert Definition. When the value of the **AlertDefinitionName** column is Null, the Alert Definition does not have a name.
UserLogonName	NVarChar(Max)	The name of the user who created the Alert Definition.
ReportURI	NVarChar(Max)	The URI that indicates which report was used to create the Alert Definition. When the value of the **ReportURI** column is Null, the ReportURI is unknown.
DataFeedURL	NVarChar(Max)	The URL to use to get the data feed for the Alert Definition.
RuleDefinition	NVarChar(Max)	The XML snippet defines the rules that are provided by the user. For more information, see section 2.1.3.5.2.
DefinitionCulture	NVarChar(Max)	The Culture Name [MSDN-NLSAPIRef] that defines the culture to use during data comparisons.
Sch_StartDate	Datetime	The start date and time when the Alert Definition recurrence starts. The time zone is assumed to be server time.
Sch_EndDate	Datetime	The end date and time when the Alert Definition recurrence ends. The time zone is assumed to be server time. When the value of the **Sch_EndDate** column is Null, no end date is specified.
Sch_RecurrenceType	Int	An enumeration that defines the recurrence type. For more information, see section 2.1.3.5.3.
Sch_RecurrenceInterval	Int	A bitmask that defines the recurrence interval. For more information, see section 2.1.3.5.3.
Attributes	Int	A bitmask that defines attributes for the Alert Definition. See section 2.1.3.5.1.1.
EmailRecipients	NVarChar(Max)	A semicolon-delimited list of email addresses to which to send alert messages.
EmailSubject	Nvarchar(Max)	The text to place in the email subject. When the value of the **EmailSubject** column is Null, no subject is specified for the Alert Definition.
EmailDescription	Nvarchar(Max)	The text to place in the email body. When the value of the **EmailDescription** column is Null, no description is specified for the Alert Definition.

2.1.3.5.1 Alert Definition Metadata

2.1.3.5.1.1 Attributes

The **Attributes** structure defines attribute information for an Alert Definition.

0	1	2	3	4	5	6	7	8
SendMessageOnResultChanges	Reserved							

SendMessageOnResultChanges (1 byte): A byte that indicates that alert messages are sent only if the alert result changes.

Value	Meaning
0	False. An alert message is sent regardless of whether the alert result changes.
1	True. An alert message is sent when the alert result changes.

2.1.3.5.2 Rule Definition

The **RuleDefinition** column contains an XML snippet that defines the rule. The following sections define the XML snippet. Appendix A shows an XSD.

The top-level element is **data-condition**.

Element	Cardinality	Type	Description
data-condition	1	Element	Root node for the data condition.

2.1.3.5.2.1 data-condition Element

Attributes

Element	Cardinality	Type	Description
Xmlns	1	String	A placeholder namespace that uniquely identifies the RuleDefiniton XML schema.

Elements

Element	Cardinality	Type	Description
scope	1	String	Values: Any, None Any. If any rows in the data feed match the data-condition, an alert instance is created. None. If no rows in the data feed match the data-condition, an alert instance is created.
clause	0-N	Element	An element that contains a set of rules to evaluate.

2.1.3.5.2.2 clause Element

Elements

Element	Cardinality	Type	Description
is-or-clause	1	Boolean	Values: True, False True. This clause is OR'ed with the previous clause. False. This clause is AND'ed with the previous clause. Clauses with an **is-or-clause** that is equal to True take precedence during evaluation.
expression-type	1	String	Values: Numeric, DateTime, String Defines the data type of the value input that is provided by the **right-operand** element **value** element. Defines the data type of the field that is provided by the **left-operand** string.
left-operand	1	String	The name of column that is defined by the data feed.
operator	1	String	The name of the operator to use to compare the value of the **left-operand** against the value of the **right-operand**. For the list of operators, see section 2.1.3.5.4.
right-operand	1	Element	The threshold to compare the **left-operand** against.

2.1.3.5.2.3 right-operand Element

Elements

Element	Cardinality	Type	Description
operand-type	1	String	Values: Value, FieldReference Value. When specified, the **value** element contains the value of a static threshold. FieldReference. When specified, the **value** element contains the column name that is defined by the data feed that provides the threshold for the clause.
value	1	String	CDATA section that contains the value as defined in [XML1.0/5] section 2.7.

2.1.3.5.3 Schedules

Alert Definitions are processed at each occurrence of a schedule. Schedules are defined as a recurrence. The recurrence starts on the Datetime that is contained in the **Sch_StartDate** column. Schedules occur at each interval offset starting at the next available interval.

If the value of **Sch_StartDate** is in the future, the Alert Definition processing starts at that date and time. If the value of **Sch_StartDate** is in the past, the Alert Definition processing starts at the next interval offset relative to the date and time.

For example, if now is 1/1/2011 07:01:00 and StartDate is 1/1/2011 06:05:00 and given an interval of 60 minutes, the first occurrence of the schedule is at 1/1/2011 07:05:00 and the next occurrence is each subsequent 60 minutes.

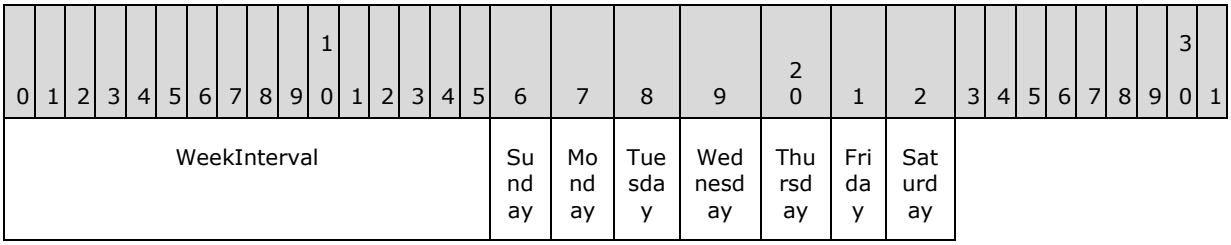
Sch_RecurrenceType	Description
1	Schedule occurs once.
2	Schedule occurs at the minutes as defined in **Sch_RecurrenceInterval**.
3	Schedule occurs at the hours as defined in **Sch_RecurrenceInterval**.
4	Schedule occurs weekly on a particular day of week as defined in **Sch_RecurrenceInterval**.

The value of the **Sch_ RecurrenceInterval** column is interpreted based on the value of **Sch_RecurrenceType** column as follows:

Sch_RecurrenceType	Sch_RecurrenceInterval Description
1	Not used.
2	Integer that represents the number of minutes between schedule occurrences.
3	Integer that represents the number of 60-minute intervals between Alert Definition processing.
4	The **Sch_ RecurrenceInterval** structure defines the number of weeks between schedule occurrences and the day of the week on which the occurrence happens. See section 2.1.3.5.3.1 for more information.

2.1.3.5.3.1 Sch_RecurrenceInterval Packet Diagram

The **Sch_ RecurrenceInterval** structure defines the number of weeks between schedule occurrences and the day of the week on which the occurrence happens.



WeekInterval (2 bytes): An integer that indicates the number of weeks to wait between schedule occurrences.

Sunday (1 bit): A bit that indicates that the schedule occurrence happens on Sunday.

Monday (1 bit): A bit that indicates that the schedule occurrence happens on Monday.

Tuesday (1 bit): A bit that indicates that the schedule occurrence happens on Tuesday.

Wednesday (1 bit): A bit that indicates that the schedule occurrence happens on Wednesday.

Thursday (1 bit): A bit that indicates that the schedule occurrence happens on Thursday.

Friday (1 bit): A bit that indicates that the schedule occurrence happens on Friday.

Saturday (1 bit): A bit that indicates that the schedule occurrence happens on Saturday.

2.1.3.5.4 Operators

The following table defines the operator names and their meaning.

Value	Description
NotSupported	The clause is omitted from the rule.
Equal	The value of the **left-operand** element equals the value that is provided by the **right-operand** element.
NotEqual	The value of the **left-operand** does not equal the value that is provided by the **right-operand**.
Less	The value of the **left-operand** is less than the value that is provided by the **right-operand**.
LessEqual	The value of the **left-operand** is less than or equal to the value that is provided by the **right-operand**.
Greater	The value of the **left-operand** is greater than the value that is provided by the **right-operand**.
GreaterEqual	The value of the **left-operand** is greater than or equal to the value that is provided by the **right-operand**.
Contains	The value of the **left-operand** contains the string that is provided by the **right-operand**.

3 Appendix A: Rule Definition XSD

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/rsalerting/2011/03/alertdefinition"
xmlns:mstns="http://schemas.microsoft.com/rsalerting/2011/03/alertdefinition"
xmlns="http://schemas.microsoft.com/rsalerting/2011/03/alertdefinition"
xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
elementFormDefault="qualified">
  <xs:element name="data-condition">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="scope" type="xs:string" minOccurs="1" />
        <xs:element name="clause" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="is-or-clause" type="xs:string" minOccurs="1" maxOccurs="1"/>
              <xs:element name="expression-type" type="xs:string" minOccurs="1"
maxOccurs="1"/>
              <xs:element name="left-operand" type="xs:string" minOccurs="1" maxOccurs="1"/>
              <xs:element name="operator" type="xs:string" minOccurs="1" maxOccurs="1"/>
              <xs:element name="right-operand" minOccurs="1" maxOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="operand-type" type="xs:string" minOccurs="1"
maxOccurs="1"/>
                    <xs:element name="field-reference" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element name="value" type="xs:string" minOccurs="0" maxOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

5 Index