

Excerpt from Callflow Software's Developer Guide

Writing Alert Rule Conditions

Alert rule conditions are Boolean functions that determine whether an alert should be turned on or off. The conditions are written in VB.Net, which is both a powerful and flexible tool. You don't have to be an expert programmer to write conditions since most common tasks are easy to accomplish.

The condition text set on the alert rule form is in affect the inner part of a VB function, which returns a Boolean value. The actual VB function called is shown

```
Namespace QFlow.Herald.Condition
Class RuleCondition
    Function Evaluate (ByRef state As Object) As Boolean
        With state.MonitoredObject

            [condition text goes here]

        End With
    End Function
End Class
End Namespace
```

The function must always return a Boolean value which represents the condition value. The state parameter passed to the function is an object of type **ConditionState**. The object contains information about the current object and the alert rule being evaluated. Code may be used to alter the **ContextParameters** dictionary in the state object. The entries in this dictionary may later be used as dynamic text parameters in the on and off messages.

Example: Monitoring the longest waiting time in services

- Set the **Monitored Object** to Service
- Set **Trigger** to Condition
- Set Condition to:

```
Return (.MaxWT > 30)
```

This condition resolves to true if the service's **Max. Waiting Time** is longer than 30 seconds.



Note - You must always use the **Return** command at the end of the function to return the Boolean value of the condition. Note that writing **.MaxWT** is equivalent to **writingstate.MonitoredObject.MaxWT**. Get the complete list of supported parameters for each monitored object type [here](#).

Example: Detecting Agent Long Service Time

- Set the **Monitored Object** to User
- Set **Trigger** to Condition
- Set **Condition** to:



```
If .Status = 3 And .TimeInStatus > 30 Then  
    Return True  
Else  
    Return False  
End If
```

This condition resolves to true if the user status is In-Service (3) and time in status is longer than 30 seconds.

Using Custom Stored Procedures

Custom stored procedures may be used to extend the capabilities of the alerts module in the following scenarios:

- You need to evaluate data not exposed by the state object
- You wish to base conditions on objects other than unit, service, and user
- You wish to implement a complex business logic
- You wish to retrieve data not exposed by the state object and use it in the condition or in messages

Custom procedures are executed before the condition is tested. The procedure should comply with the following standard:

```
CREATE PROCEDURE qfh.ProcedureName  
    @ObjectId INT
```

AS

```
SELECT TOP 1 *  
FROM Dummy
```

```
RETURN 0
```

The input parameter **@ObjectId** represents the ID of the monitored object (unit, service, or user). If the monitored object is System, the **@ObjectId** is zero.

The value returned from the procedure is stored in the **SPReturnCode** property of the state object.

The procedure may return a single-row result set. The result set is stored in the **ContextParameters** dictionary of the state object (the column name is used as the dictionary key).

