```
maintenance:
    title:          Provide notification if SIRS definition is met;;
    mlmname:        SIRS-Notification4;;
    arden:          version 2.9;;
    version:        0.9;;
    institution:    Medexter Healthcare, Vienna, Austria;;
    author:         knowledge engineering group;;
    specialist:     clinical specialists group;;
    date:           2016-08-19;;
    validation:     testing;;
library:
        purpose: ;;
        explanation: ;;
        keywords: ;;
        citations: ;;
        links: ;;
knowledge:
        type: data_driven;;
        data:
        testID:= Argument;
        Patient := object [temperature, heartRate, respRate, PaCO2,
WBcellCount, immatureBand];

        allValues:= READ as Patient {SELECT temperature, heartRate,
respRate, PaCO2, WBcellCount, immatureBand, Date as PrimaryTime
            FROM sirsvalues2 WHERE IDPatient = testID ORDER BY
Date};

        TempUpperLimit          := 38;
        TempLowerLimit          := 36;
        HeartRateUpperLimit     := 90;
        RespRateUpperLimit      := 20;
        PaCO2LowerLimit         := 32;
        WBcellCountUpperLimit   := 12000;
        WBcellCountLowerLimit   := 4000;
        ImmatureBandUpperLimit  := 10;
        ;;
        priority: ;;
        evoke: ;;
        logic:
        latestValue := latest allValues;
        temperature     := latestValue.temperature;
        heartRate       := latestValue.heartRate;
        respRate        := latestValue.respRate;
        PaCO2           := latestValue.PaCO2;
        WBcellCount     := latestValue.WBcellCount;
        immatureBand    := latestValue.immatureBand;

        //Start - Checking SIRS criteria
        counter:=0;

        if temperature is greater than TempUpperLimit or temperature
is less than TempLowerLimit then
            counter:= counter + 1;
```

```
        endif;

        if heartRate is greater than HeartRateUpperLimit then
            counter:= counter + 1;
        endif;

        if respRate is greater than RespRateUpperLimit or PaCO2 is
less than PaCO2LowerLimit then
            counter:= counter + 1;
        endif;

        if WBcellCount  is greater than WBcellCountUpperLimit or
WBcellCount is less than WBcellCountLowerLimit
        or immatureBand is greater than ImmatureBandUpperLimit then
            counter:= counter + 1;
        endif;

        if counter is greater than or equal 2 then
            notification:= localized 'SIRS';
            //End - Checking SIRS criteria

            //Start - Checking for overalerting
            previousAlert:=0;
            previousCounter:=0;
            previousValues := allValues where time of it is less
than time of latestValue;
            previousValues := previousValues where time of it is
greater than time of latestValue - 24 hours;

            for values in previousValues do
                temperature     := values.temperature;
                heartRate       := values.heartRate;
                respRate        := values.respRate;
                PaCO2           := values.PaCO2;
                WBcellCount     := values.WBcellCount;
                immatureBand    := values.immatureBand;

                if temperature is greater than TempUpperLimit or
temperature is less than TempLowerLimit then
                    previousCounter:= previousCounter + 1;
                endif;

                if heartRate is greater than HeartRateUpperLimit
then
                    previousCounter:= previousCounter + 1;
                endif;

                if respRate is greater than RespRateUpperLimit or
PaCO2 is less than PaCO2LowerLimit then
                    previousCounter:= previousCounter + 1;
                endif;

                if WBcellCount  is greater than
WBcellCountUpperLimit or WBcellCount is less than
```

```
WBcellCountLowerLimit
                or immatureBand is greater than
ImmatureBandUpperLimit then
                    previousCounter:= previousCounter + 1;
                endif;

                if previousCounter is greater than or equal 2 then
                    previousAlert:=1;
                endif;
                previousCounter:=0;
            enddo;

            if previousAlert is equal 0 then
             conclude true;
            endif;

        endif;
        //End - Checking for overalerting
        ;;
        action:
        return notification;
        ;;
        urgency: ;;

resources:
        default: en;;
        language: en
       'SIRS': "Alert for SIRS";
    ;;
    language: de
       'SIRS': "Alert f¸r SIRS";
    ;;
end:
```