**MULTIPLE SITE DOWN**

**Description:**

BTS sites are linked via microwave connections. They are configured in both hub and spoke combinations. When multiple BTS downs are detected for root TTs should be generated, after a specified wait time to allow for the events to self clear, indicating the source of the error along the microwave connection. A check needs to be made before raising the TT if the existing TT is for partial site down.

This can be achieved through the use of a parent child relationship for both the hub and spoke configurations, as long as the connectivity information is stored in a suitable place for the Impact policy to gather the connectivity data from.

The parent child event relationships, and the filters that should be used to identify the correct relationships between events and effected devices, needs to be created as this is referenced by the policy to establish the relationships. The parent child events need to be able to gather data to allow them to correlate events against each other from different devices. So if Device A is connected to device B on port X then this connectivity information needs to be looked up, to enrich the data into the event so that the parent child relationships can be established. This data needs to be gathered by Mobilink for this policy.

Source Events: ???

**EventReader Filter: ???**

**Data Type**:

SITETOPOLOGY

Status

**ImpactFlag**

\* 0 –

\* 1 –

\* 2 -

\* 3 –

\* 4 -

\* 5 -

**Pre- requisites**

ImpactFlag = ???

LinkNumber, LinkSeqNumber and EventID must be enrich in the ML\_Enrichment policy.

**Output Flag**

**ParentChild Indicator Field Values - Note these are the same as for Parentchild Policies**

\* 0 - Event not involved in Parent/Child Relationship

\* 1 - Potential Child Event

\* 2 - Potential Parent Event

\* 3 - Potential Child and Parent Event

\* 4 - Established Child Event

\* 5 - Established Parent Event

\* 6 - Child Event but no Parent Found

\* 7 - Parent Event but no Child Found

\* 8 - Relationship not found in table

**ImpactFlag**

The ImpactFlag is set to 5 only for parent events and events that don’t have any parent child relationship.

**Multiple Site Down Flow Chart**



**Policy Tree View**



**Policy Code**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\*  MultipleSiteDown.ipl  
\*  
\*  Innovise ESM Impact policy  Multiple Site Down Correlation  
\*  Chris Janes of Innovise ESM  
\*  
\*  V0.0    20100809        Chris Janes\*  
\*  v1.0    20100827        Alex Silva \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\* Here we set up static data to be used in the Policy  
\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\*  Parent / Child Indicator Field Values - Note these are the same as for Parentchild Policies  
\*  
\*      0 - Event not involved in Parent/Child Relationship  
\*      1 - Potential Child Event  
\*      2 - Potential Parent Event  
\*      3 - Potential Child and Parent Event  
\*      4 - Established Child Event  
\*      5 - Established Parent Event  
\*      6 - Child Event but no Parent Found  
\*      7 - Parent Event but no Child Found  
\*      8 - Relationship not found in table  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\* Datatypes required for this Policy  
\*  
\*   Status - alerts.status  
\*    NetworkTopology  
\*  
\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\* Here the code starts for real  
\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  
Log("MultipleSiteDown: Starting...");  
// Enrich with Network Info  
Site = Substring(@Node,0,7);  
log("Site = " + Site);  
PossParentDataType = "SITETOPOLOGY";  
// Filter = "AENDSITE = '" + Site+ "'";  // this needs to be setup  
PossParentFilter = "LINKNUMBER = "+@LinkNumber+" and SEQUENCENUMBER <= "+@LinkSeqNumber+"";  
log("PossParentFilter = " + PossParentFilter );  
CountOnly = False;  
PossLinks= GetByFilter(PossParentDataType, PossParentFilter, CountOnly);  
NumPossLinks=  Length(PossLinks);  
log("NumPossLinks for site = " + NumPossLinks);  
  
  
if(int(NumPossLinks) == 0)  
{  
    @ParentChild = 8; // Relationship not found  
    ReturnEvent(EventContainer);  
    Exit();  
}  
else  
{  
    // Possible Child and Parent  
    @ParentChild = 3;    
   ReturnEvent(EventContainer);  
   
  
    // Check for Possible Parents - Parents are defined as on this link with a Lower Seq Number (ie closer to the core)  
    PossParentDataType = "Status";    
     PossParentFilter = "LinkNumber = " +@LinkNumber+ " and LinkSeqNumber < " +@LinkSeqNumber+ " and EventId like 'MSD'";   
    CountOnly = False;  
    PossParents= GetByFilter(PossParentDataType, PossParentFilter, CountOnly);   
    NumPossParents= Length(PossParents);  
   
    if(int(NumPossParents) > 0)  
    {  
         log("Child Event");        
         @ParentChild = 4;   
  
        // Check if the Parent events is Already Processed  
        ParentDataType = "Status";   
        ParentFilter = "LinkNumber = " +@LinkNumber+ " and LinkSeqNumber < " +@LinkSeqNumber+ " and EventId like 'MSD' and ParentChild = 5";  
        CountOnly = False;  
        Parents= GetByFilter(ParentDataType, ParentFilter, CountOnly);   
        NumParents= Length(Parents);  
  
        if (int(NumParents) > 0)  
        {  
                @ParentID = Parents[0].Node;  
        }  
        // ParentID will be set when Parent is processed  
        // @ParentID = Parents[0].Node;  
    }  
    else  
    {  
        log("Parent Event");  
        // Check for Children  
        FindChildrenDataType = "Status";   
        FindChildrenFilter = "LinkNumber = " +@LinkNumber+ " and LinkSeqNumber > " +@LinkSeqNumber+ " and EventId like 'MSD'";  // this needs to be setup  
        CountOnly = False;  
        Children= GetByFilter(FindChildrenDataType, FindChildrenFilter, CountOnly);  
        Num\_Children= Length(Children);  
        if(int(Num\_Children) > 0)  
        {  
             // update all children ParentChild=4, ParentID=@Node  
            ChildrenDataType = "Status";  
            ChildrenFilter = "LinkNumber = " + @LinkNumber+ " and LinkSeqNumber > " + @LinkSeqNumber + " and EventId like 'MSD'";  
            ChildrenUpdateExpression = "ParentID = '"+@Node + "' , ParentChild = 4 , ImpactFlag = 69"; // Established Child Event  
            BatchUpdate(ChildrenDataType, ChildrenFilter, ChildrenUpdateExpression );  
        }  
        @ParentID = @Node;  
        @ParentChild = 5;  
        @ImpactFlag = 5; // enable event to be processed by Site and Cell Down Policy  
          
    }  
    ReturnEvent(EventContainer);  
}