

IBM Tivoli Netcool/OMNibus 7.0 Interface to Tivoli Service Desk
Quick setup guide – TSD element

Steps to creating an Omnibus interface

The omnibus to TSD interface is a multi step process. First it involves adding some additional fields to the Incident form (Ticket object). These additional fields will also need to be added to the Ticket object itself using the database configuration tool. Once these fields have been added to the data capture applications, the interface objects themselves need to be created and the final stage is the creation of the interface table that will stage the inbound data between Omnibus and TSD. After the TSD elements has been configured focus moves to configuring Omnibus. Omnibus will need to have the policy added and the JDBC connections created to the TSD staging table.

1. Open the database configuration application. The search for the TICKET object – do not work on the incident or problem views. Database Views are difficult to work with when interfacing.

I would recommend adding the following fields as a minimum. The lengths have been made to match the default fields in the ObjectServer to avoid truncation.

Maximo Column Name	Type	Length	Comment
OMNI_NODE	ALN	64	This is commonly the node name
OMNI_NODEALIAS	ALN	64	This is commonly the IP or FQDN
OMNI_ALERTGROUP	ALN	255	
OMNI_ALERTKEY	ALN	255	
OMNI_SEVERITY	ALN	32	Severity as text string, converted by Impact
OMNI_SUMMARY	ALN	255	
OMNI_SERIAL	INT	12	This is required to update the Netcool event. Identifier string could also be used.
OMNI_TALLY	INT	12	
OMNI_FIRST	DATETIME	10	FirstOccurrence converted using TO_DATE in the insert
OMNI_LAST	DATETIME	10	LastOccurrence converted using TO_DATE in the insert

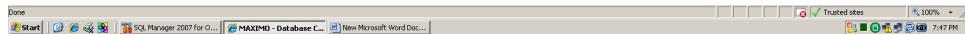
Note : Screenshots do not show the same fields.

The screenshot shows the MAXIMO Database Configuration interface. The main window displays the 'TICKET' object. On the left, there's a tree view of attributes under 'Attributes'. On the right, there are two tables: 'Attributes' and 'Description'. The 'Attributes' table lists columns like OMNI_NODE, OMNI_NODEALIAS, etc., with their types (ALN, DATE, INT, etc.) and lengths. The 'Description' table lists columns like Omn_alertgroup, Omn_first, etc., with their types (ALN, DATE, etc.) and lengths. Two sections of the interface are highlighted with large red circles: the 'Attributes' table and the 'Description' table.

Status	Attribute	Description	Type	Length	Date	Required?
	OMNI_ALERTGROUP	Omn_alertgroup	ALN	50	0	<input type="checkbox"/>
	OMNI_FIRST	Omn_first	DATE	4	0	<input type="checkbox"/>
	OMNI_LAST	Omn_last	DATE	4	0	<input type="checkbox"/>
	OMNI_LOCATION	Omn_location	ALN	64	0	<input type="checkbox"/>
	OMNI_MANAGER	Omn_manager	ALN	65	0	<input type="checkbox"/>
	OMNI_NODE	Omn_node	ALN	14	0	<input type="checkbox"/>
	OMNI_SERIAL_NUMBER	Omn_serial_number	ALN	20	0	<input type="checkbox"/>
	OMNI_SEVERITY	Omn_severity	ALN	4	0	<input type="checkbox"/>
	OMNI_SITE_INFO	Omn_site_info	ALN	100	0	<input type="checkbox"/>
	OMNI_SUMMARY	Omn_summary	ALN	255	0	<input type="checkbox"/>

Status	Column	Type	Length	Date	Required?
	Omn_alertgroup	ALN	50	0	<input type="checkbox"/>
	Omn_first	DATE	4	0	<input type="checkbox"/>
	Omn_last	DATE	4	0	<input type="checkbox"/>
	Omn_location	ALN	64	0	<input type="checkbox"/>
	Omn_manager	ALN	65	0	<input type="checkbox"/>
	Omn_node	ALN	14	0	<input type="checkbox"/>
	Omn_serial_number	ALN	20	0	<input type="checkbox"/>
	Omn_severity	ALN	4	0	<input type="checkbox"/>
	Omn_site_info	ALN	100	0	<input type="checkbox"/>
	Omn_summary	ALN	255	0	<input type="checkbox"/>

Attribute	Description	Type	Length	Scale	Secured
omn	Omnibus tally	ALN	6	0	
OMNI_TALLY	OMNI tally	ALN	6	0	



2. Save these modifications
3. Now log off Maximo and stop the Maximo web application within Websphere or BEAWeblogic. In Weblogic login as weblogic:weblogic then navigate to mydomain>Deployments>Applications>maximo and the Deploy tab. Use the 'Stop Applications' button to stop the maximo server. 'Deploy Application' can be used to restart Maximo once the database has been migrated.
4. Now open up a command prompt. Change directory to the Maximo installation directory. Then change to tools\maximo. Within this folder run configdb. If Maximo instance has stopped then configdb will run and reconfigure the DB.
5. When configdb has run it is **safe** to start Maximo – and hopefully it will start.
6. Add required Omnibus fields to the Incident application. Any or all of the attributes outlined below would be suitable. Open the application designer and add the fields using the design palate. You can select the existing information panel, copy and paste and then edit the properties for the fields to match the new OMNI_x fields.

Incident Information

Omni Serial ID: 12345	Omni Site Address:	Omni Alert Group:	Omni First Occurrence: 8/21/07
Omni NODE:	Omni Location: LONDON	Omni Manager: SVEN	OMN Tally: 12135
	Omni Summary:	Omni Severity: 2	OMN Last Occurrence: 8/22/07

User Information

Reported By:	Affected Person:
Name:	Name:
Phone:	Phone:
E-mail:	E-mail:

Incident Details

Summary: TEST 3	Classification:
Details:	Description:
Asset:	Reported Priority:
Location:	Internal Priority:
GL Account:	Service Group:
Asset Site:	Service:
	Vendor:
	Site:
	SLA Applied?

Dates

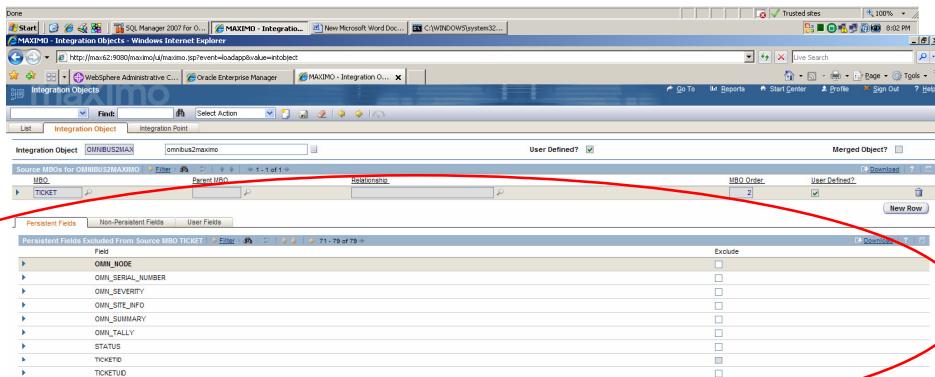
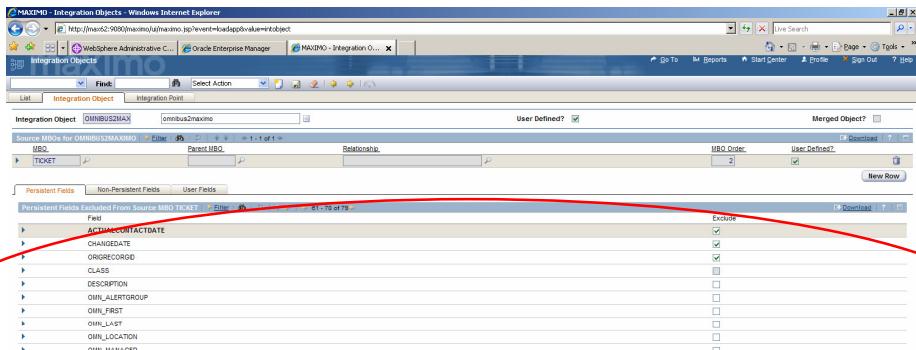
Reported Date: 8/22/07 10:13 PM	Target Contact:	Actual Contact:	Global Issues?
Affected Date:	Target Start:	Actual Start:	Related to Global ID:
	Target Finish:	Actual Finish:	Global Class:

Related Assets

- Up to this point the base attributes have been added to Maximo. Open up the INTEGRATION OBJECTS application. Create a new object called OMNIBUS2MAXIMO. Create a new MBO using the TICKET Object. Now save the record. In the Persistent Fields exclude all fields except Mandatory (CLASS, DESCRIPTION, STATUS, TICKETID, TICKETUID, ASSETNUM) and the newly added OMNI_x fields.

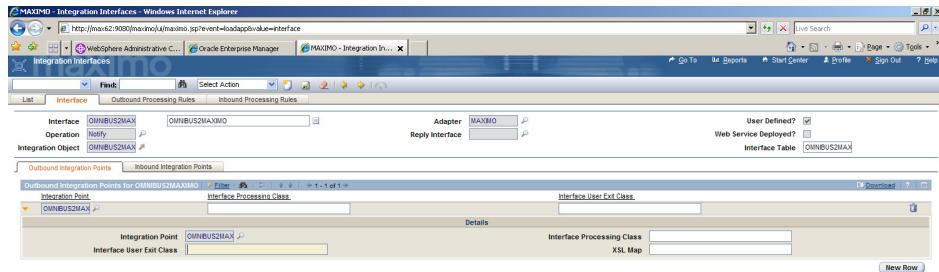
Integration Objects

Integration Object: OMNIBUS2MAXIMO	User Defined? <input checked="" type="checkbox"/>	Merged Object? <input type="checkbox"/>																						
MBO: TICKET	Parent MBO:	Relationship:																						
Persistent Fields	Non-Persistent Fields	User Fields																						
Persistent Fields Extracted from Source MBO TICKET <table border="1"> <thead> <tr> <th>Field</th> <th>Exclude</th> </tr> </thead> <tbody> <tr><td>SITEVMT</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>ISGLOBAL</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>EXTERNALREQ</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>COMMODITYGROUP</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>STATUSDATE</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>CLASSTRUCTURED</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>PROBLEMCODE</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>REPORTEDDSTRTY</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>ORGID</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>GLOBALTICKETCLASS</td><td><input checked="" type="checkbox"/></td></tr> </tbody> </table>			Field	Exclude	SITEVMT	<input checked="" type="checkbox"/>	ISGLOBAL	<input checked="" type="checkbox"/>	EXTERNALREQ	<input checked="" type="checkbox"/>	COMMODITYGROUP	<input checked="" type="checkbox"/>	STATUSDATE	<input checked="" type="checkbox"/>	CLASSTRUCTURED	<input checked="" type="checkbox"/>	PROBLEMCODE	<input checked="" type="checkbox"/>	REPORTEDDSTRTY	<input checked="" type="checkbox"/>	ORGID	<input checked="" type="checkbox"/>	GLOBALTICKETCLASS	<input checked="" type="checkbox"/>
Field	Exclude																							
SITEVMT	<input checked="" type="checkbox"/>																							
ISGLOBAL	<input checked="" type="checkbox"/>																							
EXTERNALREQ	<input checked="" type="checkbox"/>																							
COMMODITYGROUP	<input checked="" type="checkbox"/>																							
STATUSDATE	<input checked="" type="checkbox"/>																							
CLASSTRUCTURED	<input checked="" type="checkbox"/>																							
PROBLEMCODE	<input checked="" type="checkbox"/>																							
REPORTEDDSTRTY	<input checked="" type="checkbox"/>																							
ORGID	<input checked="" type="checkbox"/>																							
GLOBALTICKETCLASS	<input checked="" type="checkbox"/>																							

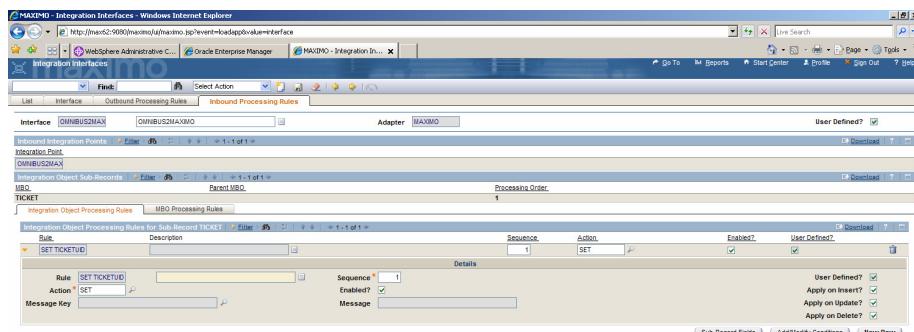


Done Start | Internet Explorer | SQL Manager 2007 for O... | MAXIMO - Integratio... | New Microsoft Word Doc... | C:\WINDOWS\system32... Trusted sites 8:02 PM

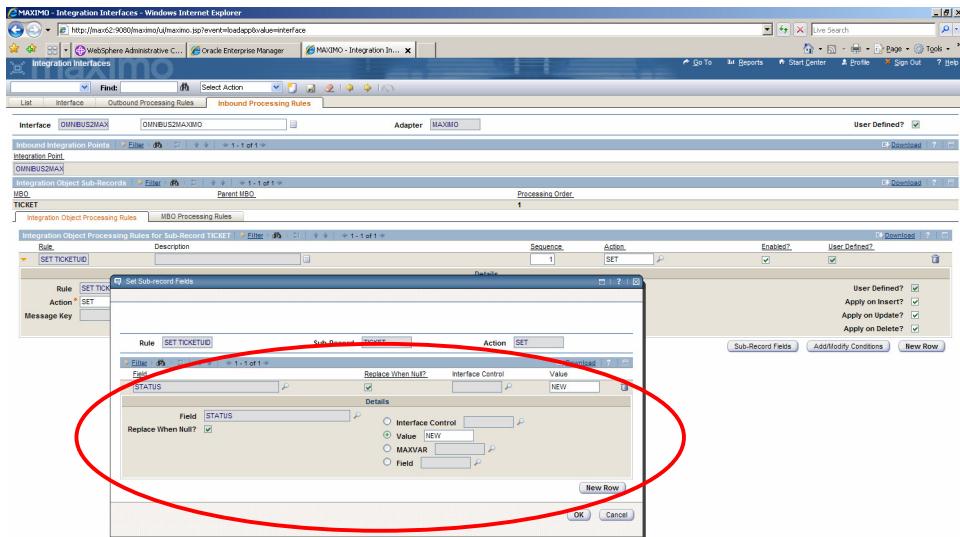
8. Create a new Integration Point for the inbound. The screen shot below shows the integration point that corresponds to the INTEGRATION INTERFACES.
9. Create a new INTERFACE object.



10. Tie up the inbound and outbound integration point to those defined in the INTERFACE OBJECTS.
11. Click on the Inbound Processing Rules tab and create an Integration Object Sub-Records MBO. Use Ticket as the MBO.
12. Now add a new rule. The reason for this is that the Status needs to be populated as information comes through from Omnibus.



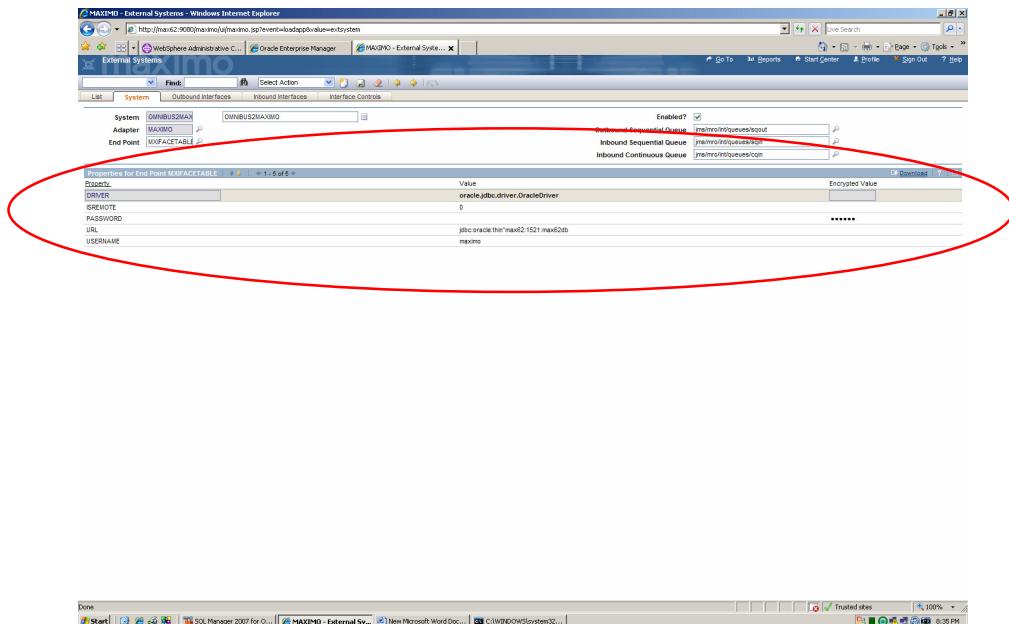
13. click on Sub-Record Fields button and add a rule that populates the status field on all new submits to the Ticket record.



14. Save the record.
15. Open up the External Systems application. Create a new System. Give the System a name, select Maximo as the Adaptor and MXINFACETABLE as the End Point.
16. Save this object.
17. Configure the Properties for End Point MXINFACETABLE. The values are:
 - a. DRIVER – VALUE - oracle.jdbc.driver.OracleDriver
 - b. ISREMOTE – VALUE - 0
 - c. PASSWORD – ENCRYPTED VALUE - maximo
 - d. URL – VALUE - jdbc:oracle:thin@max62:1521:max62db

On v6.2.1 Transportation 6.3.0 demo VM this is -

 - e. USERNAME – VALUE – maximo



18. All the values needed can be obtained from Maximo.properties as shown in the following extract:

```
// Database Schema Owner
mxe.db.schemaowner=maximo

// Specification of JDBC Driver
// e.g. SqlServer i-net opta 2000 driver
// mxe.db.driver=com.inet.tds.TdsDriver
// e.g. DB2 driver:
// mxe.db.driver=com.ibm.db2.jcc.DB2Driver
// for Oracle it is as follows...
// oracle thin driver
    mxe.db.driver=oracle.jdbc.driver.OracleDriver

// JDBC "url" of database -- varies with the particular database you're
// connecting to...
// e.g. SqlServer 6.5 i-net opta 2000 driver
//
mxe.db.url=jdbc:inetdae6:hostname:port?database=dbname&language=us_english&nowarnings=true
// e.g. SqlServer 7.0 or higher i-net opta 2000 driver
//
mxe.db.url=jdbc:inetdae7a:hostname:port?database=dbname&language=us_english&nowarnings=true
// e.g. DB2:
// mxe.db.url=jdbc:db2://localhost:50000/dbalias
// e.g Oracle thin mxe.db.url=jdbc:oracle:thin:@<HOST>:<PORT>:<SID>
// Oracle thin driver
    mxe.db.url=jdbc:oracle:thin:@max62:1521:max62db

// Database login name -- depends on driver, e.g. sometimes it's name=
// If running under SQLServer, this user must have sysadmin role,
// as defined via sp_addsrvrolemember.
// For DB2 this is an O/S user.
    mxe.db.user=maximo
```

```
// Database login password -- depends on driver, e.g. sometimes it's passwd=
// For DB2, this is the O/S password.
mxe.db.password=maximo
```

19. Click on the Inbound Interfaces tab. And create the link to the INTEGRATION INTERFACES object. Save the record.

The screenshot shows a web browser window titled "MAXIMO - External Systems - Windows Internet Explorer". The URL is "http://maxi02:9080/maximo/jsp/event=&loadappvalue=extsystem". The page displays a table of "Inbound Interfaces". A single row for "OMNIBUS2MAX" is selected, and its details are shown in a modal dialog box. The "Enabled?" checkbox is checked.

20. Now create the interface table by selecting

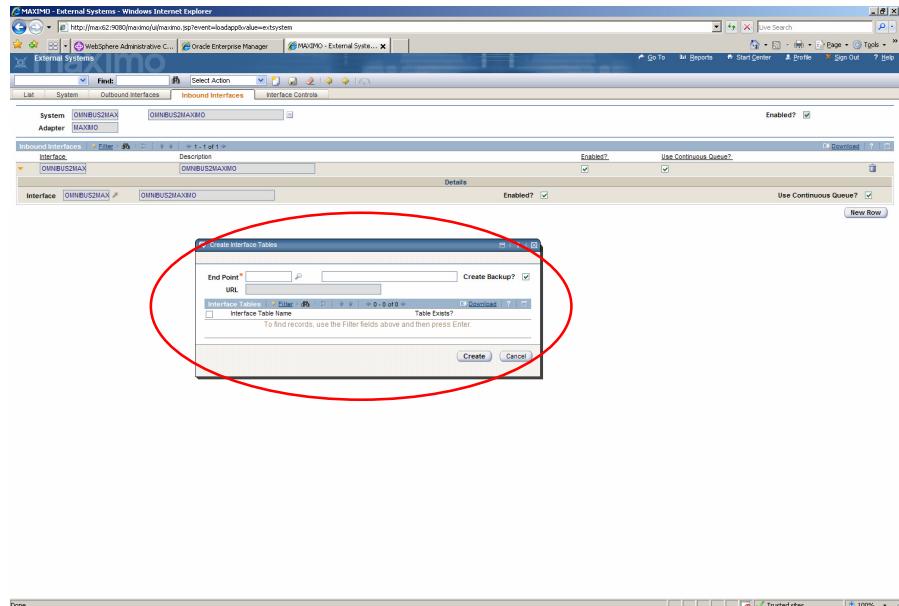
The screenshot shows the same MAXIMO - External Systems interface as before, but the "Select Action" dropdown menu is now open, revealing various administrative options. A red circle highlights this dropdown menu.

21. From the drop down menu select Create Interface Tables. This opens up the Create Interface dialogue table.

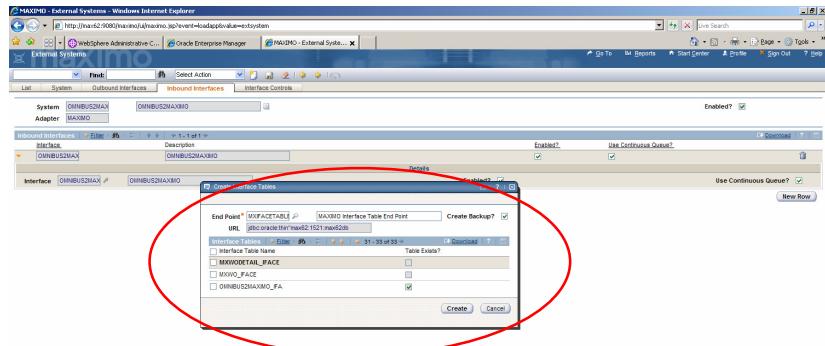
Draft version 0.1

Page 11

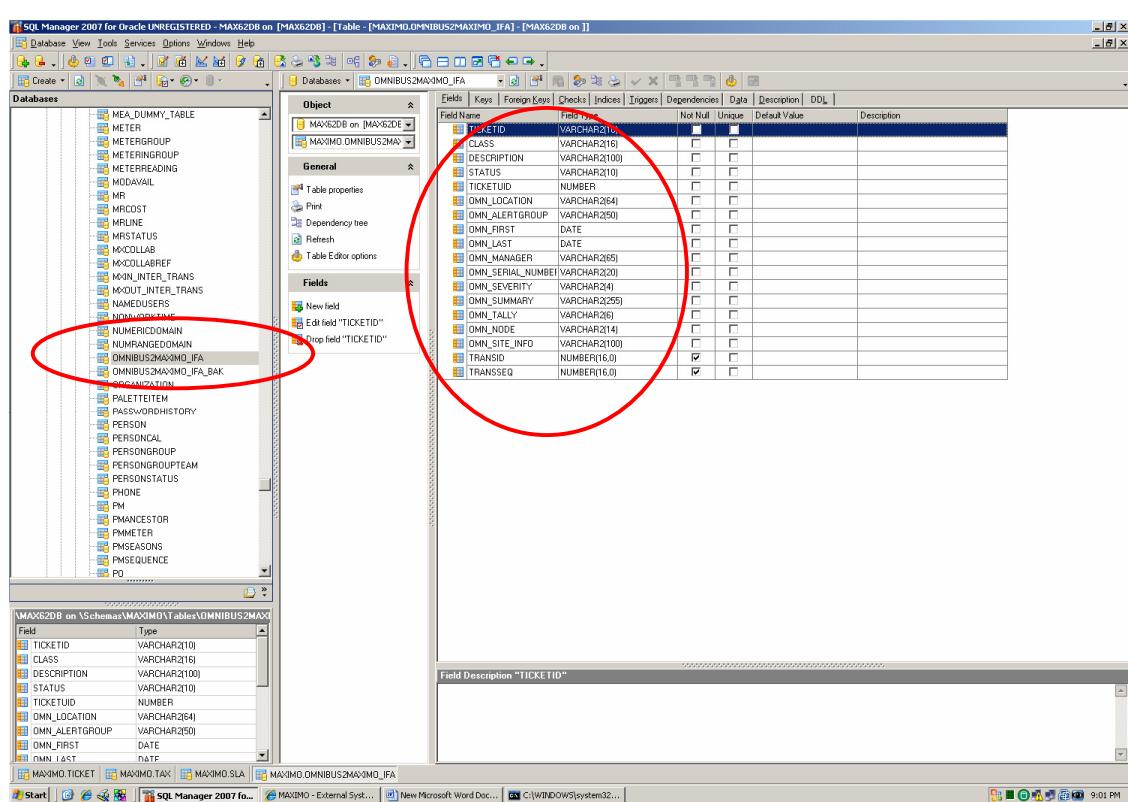
06/11/2008



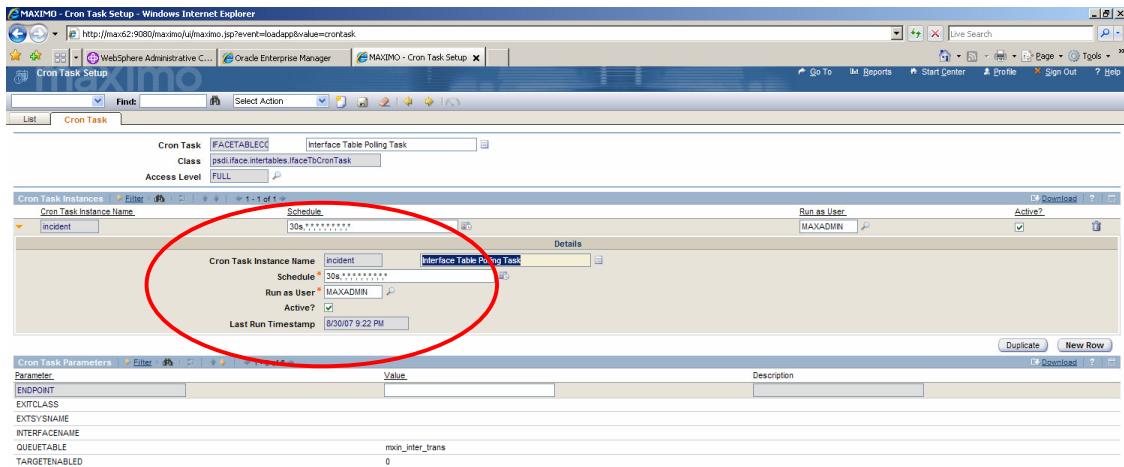
22. Select an End Point – MXIFACETABLE. Check that the values are as bellow:



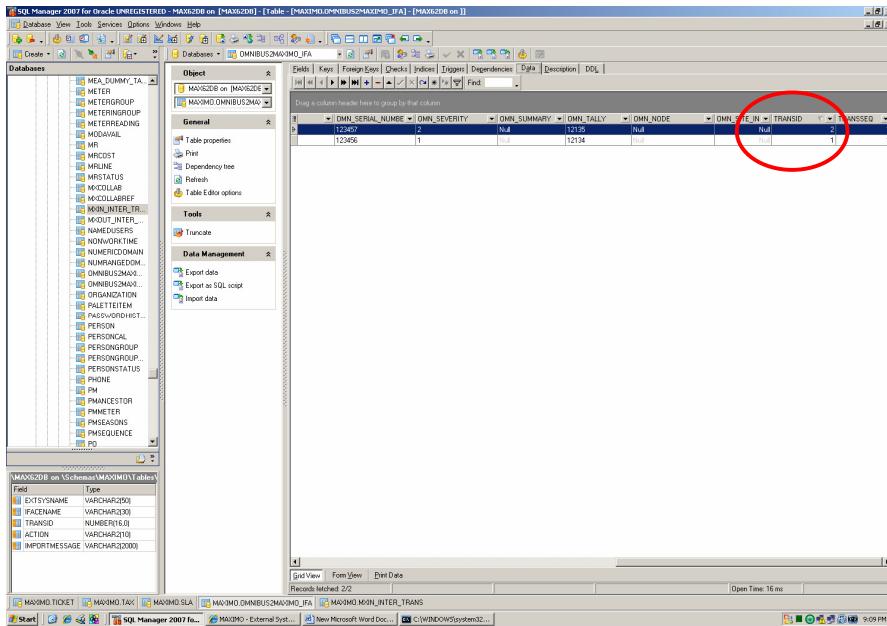
23. Select create. In the interface table exist select it and it will be updated if it does not exist you will be prompted for a name. The next step is that Maximo create the interface table. In this interface the table is called OMNIBUS2MAXIMO_IFA. The screenshot below shows the table structure.



24. Create a cron task to pick up records inputted into the OMNIBUS2MAXIMO_IFA table. Select the CRON TASK SETUP application, select the IFACETABLECONSUMER and add a new Cron Task Instance Name. Give it a 30second schedule and activate this Cron Task.

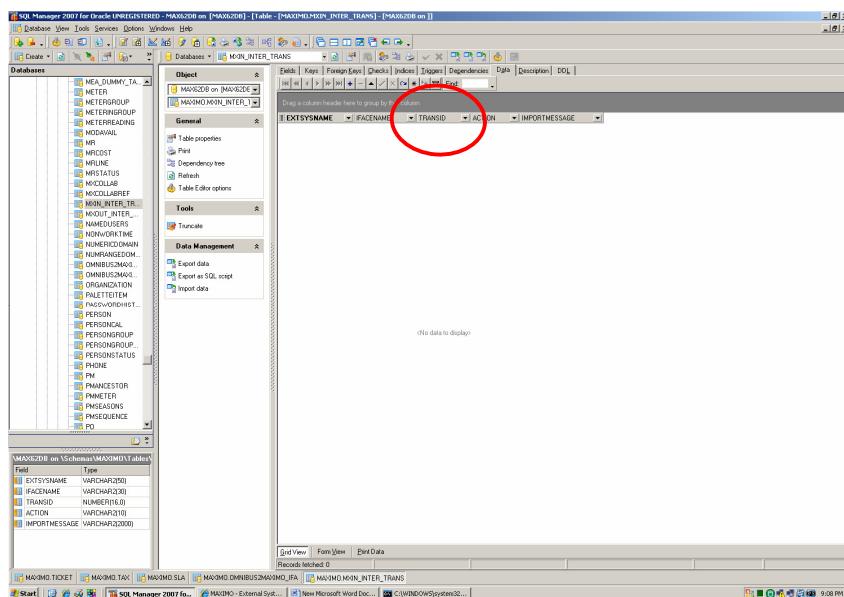


25. The last piece needed to make this interface work data needs to be added to two tables. The first is that data needs to be inserted into the OMNIBUS2MAXIMO_IFA and the second table that needs to be added to is the MXIN_INTER_TRANS table.



26. Along with the data that is inserted into the OMNIBUS2MAXIMO_IFA table, one additional piece of data is also needed. The TRANSID field needs to be incremented per entry. For each OMNIBUS2MAXIMO_IFA record for which there is a TRANSID there will need to be a corresponding value in the MXIN_INTER_TRANS table. Values are as follows:

- EXTSYSNAME – OMNIBUS2MAXIMO
- IFACENAME – OMNIBUS2MAXIMO
- TRANSID – this value needs to correspond to the same integer value in OMNIBUS2MAXIMO_IFA. For each value in this field in this table, will result in the corresponding OMNIBUS2MAXIMO_IFA record to be processed. If the TRANSID does not match there will be no processing.
- ACTION - ADDCHANGE



27. Once the crone task has run the MXIN_INTER_TRANS table is cleaned out. If there are no errors a new record that corresponds to a record in the OMNIBUS2MAXIMO_IFA will be created in the TICKET object.
28. Test the interface by Adding a row of data into the OMNIBUS2MAXIMO_IFA table and a corresponding entry into the MXIN_INTER_TRANS table. Note the TRANSID needs to be the same value in both entries. If not an error will result. When adding the row of data into the OMNIBUS2MAXIMO_IFA table use the omnibus sequence id as the TSD TICKETID with a prefix of OMNI00000000. Makes it easier to find when searching for entries in the incident application.
29. Search for the test submit in the incident application

30. Now configure Omnibus to link to OMNIBUS2MAXIMO_IFA table and create the policy to populate this table.

Impact Policies

There are two alternate methods for creating a new ticket. You can add new fields to a Data Type or you can use DirectSQL to insert directly into the Maximo tables.

Data Type Method
Create a Data Type

Create new Incident

To do this you need to create a Data source for the Maximo database. The default user is **system :manager**

Edit 'Oracle: MAXIMO' Data Source - Windows Internet Explorer

http://192.168.1.22:8080/nci/DataSourceMainServlet?action=editDataSource&dataSourceName=MAXIMO

* required field

General Settings:

Data Source Name: * required field

Username:

Password:

Maximum SQL Connection:

Database Failure Policy:

Fail over
 Fail back
 Disable Backup

Primary Source:

Host Name:

Port:

SID:

Backup Source:

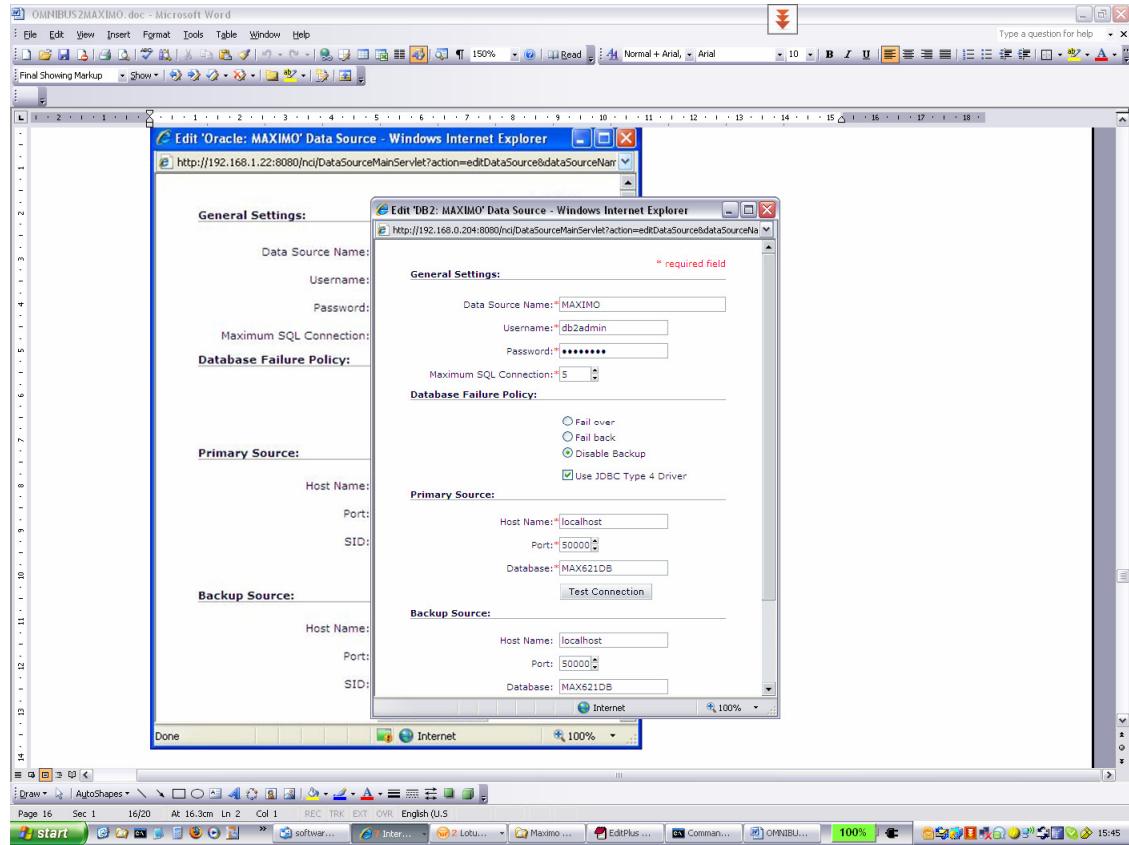
Host Name:

Port:

SID:

Done

Internet 100%



Edit 'DB2: MAXIMO' Data Source - Windows Internet Explorer

http://192.168.0.204:8080/nci/DataSourceMainServlet?action=editDataSource&dataSourceNa

* required field

General Settings:

Data Source Name: * required field

Username: * required field

Password: * required field

Maximum SQL Connection: * required field

Database Failure Policy:

Fail over
 Fail back
 Disable Backup

Use JDBC Type 4 Driver

Primary Source:

Host Name: * required field

Port: * required field

Database: * required field

Backup Source:

Host Name:

Port:

Database:

Internet 100%  

The example policy uses a DataType to insert the new information into the OMNIBUS2MAXIMO_IFA and MXIN_INTER_TRANS tables.
The DataType definitions are

Maximo MXIN_INTER_TRANS table DataType

The screenshot shows the NETCOOL Suite interface in a web browser. The left sidebar contains navigation links for Projects, Global, Data Sources, Data Types, Operator Views, Policies, and Service Status. The main panel is titled "Maximo_CreateIncident_datatype" and shows the configuration for a "Data Type".

General Settings:

- Data Type Name: Maximo_INTER
- Data Source Name: MAXIMO
- Display Icon: Browse...
- State: Enabled

Table Description:

- Base Table: MXIN_INTER_TRANS
- New Field:

Fields:

Select:	ID	Field Name	Format	Display Name	Description	Key Field	Move	Edit
<input type="checkbox"/>	EXTSYSNAME	EXTSYSNAME	STRING	EXTSYSNAME	EXTSYSNAME	<input type="checkbox"/>	<input type="button" value="Move"/>	<input type="button" value="Edit"/>
<input type="checkbox"/>	IFACENAME	IFACENAME	STRING	IFACENAME	IFACENAME	<input type="checkbox"/>	<input type="button" value="Move"/>	<input type="button" value="Edit"/>
<input type="checkbox"/>	TRANSID	TRANSID	INTEGER	TRANSID	TRANSID	<input checked="" type="checkbox"/>	<input type="button" value="Move"/>	<input type="button" value="Edit"/>
<input type="checkbox"/>	ACTION	ACTION	STRING	ACTION	ACTION	<input type="checkbox"/>	<input type="button" value="Move"/>	<input type="button" value="Edit"/>
<input type="checkbox"/>	IMPORTMESSAGE	IMPORTMESSAGE	STRING	IMPORTMESSAGE	IMPORTMESSAGE	<input type="checkbox"/>	<input type="button" value="Move"/>	<input type="button" value="Edit"/>

Data Filter and Ordering:

- Filter:
- Order By:

Maximo OMNIBUS2MAXIMO_IFA Table DataType

The screenshot shows the NETCOOL Suite interface in a Windows Internet Explorer window. The title bar says "The NETCOOL Suite - Windows Internet Explorer". The address bar shows the URL "http://192.168.0.204:8080/nci/main". The main content area is titled "NETCOOL® Suite" and shows the "Data Types" section. A sub-section titled "Edit: Maximo_INTERFACE" is active. The "Data Source Name" is set to "MAXIMO". The "Table Description" section shows the "Base Table" as "OMNIBUS2MAXIMO_IFA". The "Fields" table lists various columns with their data types and descriptions:

Select:	ID	Field Name	Format	Display Name	Description	Key Field	Move	Edit
<input type="checkbox"/>	TICKETID	TICKETID	STRING	TICKETID	TICKETID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CLASS	CLASS	STRING	CLASS	CLASS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	DESCRIPTION	DESCRIPTION	STRING	DESCRIPTION	DESCRIPTION	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	STATUS	STATUS	STRING	STATUS	STATUS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	TICKETUID	TICKETUID	INTEGER	TICKETUID	TICKETUID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_ALERTGROUP	OMNI_ALERTGROUP	STRING	OMNI_ALERTGROUP	OMNI_ALERTGROUP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_SEVERITY	OMNI_SEVERITY	STRING	OMNI_SEVERITY	OMNI_SEVERITY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_NODE	OMNI_NODE	STRING	OMNI_NODE	OMNI_NODE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_LOCATION	OMNI_LOCATION	STRING	OMNI_LOCATION	OMNI_LOCATION	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_SUMMARY	OMNI_SUMMARY	STRING	OMNI_SUMMARY	OMNI_SUMMARY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_SERIAL_NUMBER	OMNI_SERIAL_NUMBER	STRING	OMNI_SERIAL_NUMBER	OMNI_SERIAL_NUMBER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_FIRST	OMNI_FIRST	DATE	OMNI_FIRST	OMNI_FIRST	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_LAST	OMNI_LAST	DATE	OMNI_LAST	OMNI_LAST	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_MANAGER	OMNI_MANAGER	STRING	OMNI_MANAGER	OMNI_MANAGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	OMNI_TALLY	OMNI_TALLY	STRING	OMNI_TALLY	OMNI_TALLY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	TRANSID	TRANSID	INTEGER	TRANSID	TRANSID	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	TRANSSEQ	TRANSSEQ	INTEGER	TRANSSEQ	TRANSSEQ	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Below the table, there is a "Data Filter and Ordering" section with a "Filter:" input field.

I have also created a Data Type to convert the severity into text and another to hold a count which is used to generate a unique Ticket reference.

Ticket ID DataType

The screenshot shows the NETCOOL Suite interface in a Windows Internet Explorer browser window. The URL is <http://192.168.0.204:8080/nci/main>. The user is logged in as 'admin'. The main menu bar includes 'Logout | Help' and the IBM logo. The left sidebar contains 'Projects' (Default selected), 'Data Sources' (CorrelationEvents, CorrelationTable, Maximo_Assets, Maximo_DevAssets, Maximo_INTER, Maximo_Interface, Maximo_Tickets, NCMS_Alerts_Status, SeverityConversion, TicketID, WebtopLog_Topo, WebtopXMLTopology), 'Operator Views' (CommandExecutionManager, CommandLineManager, CorbaNameService, DatabaseListener, DefaultEmailReader, DefaultEventReader, DefaultFileReader, DefaultJobberReader, DefaultPolicyActivator, EmailSender, EventProcessor, Event_Correlation_Service, HibernationPolicyActivator), and 'Policies'. The central workspace shows the 'Data Types' configuration for 'TicketID'. The 'General Settings' section includes 'Data Type Name: TicketID' (marked as required), 'Display Icon: Browse...', and 'State: Persistent'. The 'Additional Fields' section shows a table with one row:

Select:	ID	Field Name	Format	Display Name	Description	Move	Edit
<input type="checkbox"/> (all)	TicketID	TicketID	INTEGER	TicketID			

Below the table are 'New Field' and 'Delete' buttons, and a 'Display Name Field:' dropdown set to 'KEY'. The bottom status bar shows 'Done', 'Internet', and '100%'.

Severity Conversion DataType

The screenshots show the configuration of a data type in the NETCOOL Suite. The top screenshot shows the initial creation of the data type, while the bottom screenshot shows the list of data items for this type.

Top Screenshot: Data Type Creation

The 'Data Types' section shows a list of existing data types. A new data type, 'SeverityConversion', is being created. The 'General Settings' panel shows:

- Data Type Name: SeverityConversion
- Display Icon: (radio button selected)
- State: Persistent

The 'Additional Fields' panel shows a single field named 'Severity' with type STRING and description Severity.

Select:	ID	Field	Format	Display Name	Description	Move	Edit
(all)		Severity	Severity	Severity	Severity		

Bottom Screenshot: Data Item Management

The 'Data Items' section lists the objects for the 'SeverityConversion' data type. There are 6 objects:

Selected:	KEY	Severity	Links	Edit
<input checked="" type="checkbox"/>	5	Critical		
<input checked="" type="checkbox"/>	4	Major		
<input checked="" type="checkbox"/>	3	Warning		
<input checked="" type="checkbox"/>	2	Information		
<input checked="" type="checkbox"/>	1	Indeterminate		
<input checked="" type="checkbox"/>	0	Clear		
Delete				

Additional fields in the ObjectServer were created for Maximo status information and for correlation purposes.

<input type="checkbox"/>	MaximoStatus	MaximoStatus	INTEGER	MaximoStatus	MaximoStatus	<input type="checkbox"/>		
<input type="checkbox"/>	MaximoTicketID	MaximoTicketID	LONG_STRING	MaximoTicketID	MaximoTicketID	<input type="checkbox"/>		
<input type="checkbox"/>	MaximoAssetNum	MaximoAssetNum	LONG_STRING	MaximoAssetNum	MaximoAssetNum	<input type="checkbox"/>		
<input type="checkbox"/>	MaximoTicketUID	MaximoTicketUID	LONG_STRING	MaximoTicketUID	MaximoTicketUID	<input type="checkbox"/>		
<input type="checkbox"/>	MaximoRequestType	MaximoRequestType	LONG_STRING	MaximoRequestType	MaximoRequestType	<input type="checkbox"/>		
<input type="checkbox"/>	MaximoTimestamp	MaximoTimestamp	INTEGER	MaximoTimestamp	MaximoTimestamp	<input type="checkbox"/>		
<input type="checkbox"/>	MaximoTicketStatus	MaximoTicketStatus	LONG_STRING	MaximoTicketStatus	MaximoTicketStatus	<input type="checkbox"/>		

Create New Incident Policy using DataType on DB2

```
// Policy creates incident in Maximo by inserting records into the Maximo staging table
// It then queries the TICKET table to find the ticket reference

// Use event reader to trigger policy based on @Grade=1 set by a tool
// policy updates @MaximoStatus=2 when complete

Log("Entering CreateIncident for "+@Summary);
if(EventContainer.MaximoTicketID<>""){
    Exit();
}

LogCount = 1;
Log("Maximo_CreateIncident :" + LogCount + ": New event received from ObjectServer");
LogCount = int(LogCount) + 1;

// Get next ticket ID from internal datatype and increment value
Results = GetByKey('TicketID',1,1);
NextTicketID = Int(Results[0].TicketID) + 1;
NewTicketID = NewObject();
NewTicketID.TicketID = NextTicketID;
NewTicketID.KEY = 1;
AddDataItem('TicketID',NewTicketID);

NewTicket = NewObject();
NewInter = NewObject();

// Assign values we need from the event

NewTicket.TICKETID = 'OMNI'+NextTicketID;

// Get the type of request from the Event rather than hard coded
// NewTicket.CLASS = 'INCIDENT';
if(EventContainer.MaximoRequestType <> ""){
    NewTicket.CLASS = EventContainer.MaximoRequestType;
} else {
    Log("Maximo_CreateIncident :" + LogCount + ": No Ticket Type received from ObjectServer, so assuming INCIDENT!!!");
    LogCount = int(LogCount) + 1;
    NewTicket.CLASS = EventContainer.MaximoRequestType;
    // Exit ();
};

NewTicket.STATUS = "";
NewTicket.TICKETUID = 0;
NewTicket.OMNI_NODE = EventContainer.Node;
```

```
NewTicket.OMNI_ALERTGROUP = EventContainer.AlertGroup;
NewTicket.OMNI_SERIAL_NUMBER = EventContainer.ServerSerial;

// Lookup conversion from severity value to name
Results = GetByKey('SeverityConversion', EventContainer.Severity, 1);
NewTicket.OMNI_SEVERITY = Results[0].Severity;

//NewTicket.ASSETNUM = EventContainer.MaximoAssetNum;
NewTicket.OMNI_LOCATION = EventContainer.Location;
NewTicket.OMNI_SUMMARY = EventContainer.Summary+" : "+EventContainer.MaximoAssetNum;
NewTicket.DESCRIPTION = "";
NewTicket.OMNI_TALLY = EventContainer.Tally;
NewTicket.OMNI_MANAGER = EventContainer.Manager;
NewTicket.OMNI_FIRST = LocalTime(EventContainer.FirstOccurrence, 'yyyy-MM-dd HH:mm:ss.S');
NewTicket.OMNI_LAST = LocalTime(EventContainer.LastOccurrence, 'yyyy-MM-dd HH:mm:ss.S');
NewTicket.TRANSID = Int(NextTicketID);
NewTicket.TRANSSEQ = 1;

Log("Maximo_CreateIncident :" + LogCount+": Data " + NewTicket);
LogCount = int(LogCount) + 1;

AddDataItem('Maximo_Interface', NewTicket);

NewInter.TRANSID = NewTicket.TRANSID;
NewInter.EXTSYSNAME = 'NETCOOL';
NewInter.IFACENAME = 'OMNIBUS2MAXIMO';
NewInter.ACTION = 'AddChange';

AddDataItem('Maximo_INTER', NewInter);

//Log("Maximo_CreateIncident :" + LogCount+": Data " + NewInter);
//LogCount = int(LogCount) + 1;

//Log("Maximo_CreateIncident - eventcontainer: "+EventContainer.Node+ " - Serial: " +
EventContainer.Serial);

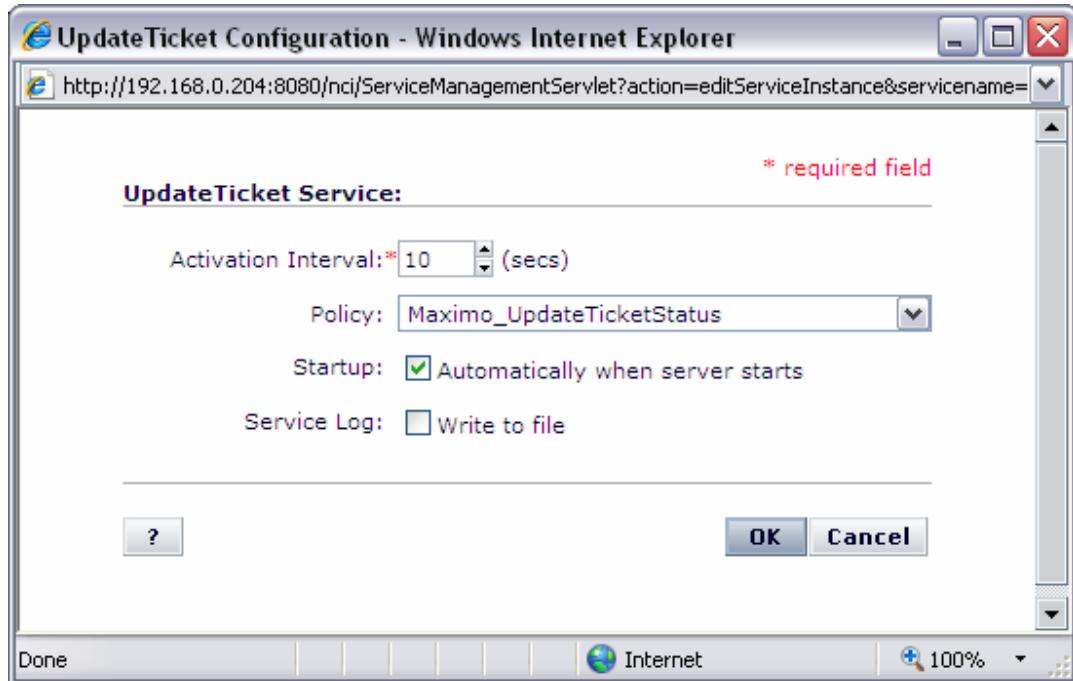
// Update the MaximoStatus field so that we only process once
// Put the new TicketID into the event
@MaximoStatus = 2;
@MaximoTicketID = NewTicket.TICKETID;
@MaximoTimestamp = GetDate();
ReturnEvent(EventContainer);
```

Create New Incident via DirectSQL on Oracle

If you are using an Oracle database you will find that the dataType method is not possible as the database does not autoincrement the TICKETSEQ as in the previous example. This means that Impact has to use the DirectSQL method so that it can create an INSERT statement that uses TICKETSEQ.nextval to insert the sequence number.

Update Ticket Status Policy

The purpose of this policy is to check whether there are any status changes to tickets that have been raised. It does this by checking the Netcool events for those which have had a ticket raised and then checks whether the status in Maximo is different, and if so it updates the Netcool event. This is executed on a timed basis as a service.



```

// Policy that monitors open tickets and updates the event status to reflect the TICKET table

// This policy is run every 10 seconds by a policy activator service

LogCount = 1;
log("Maximo_UpdateTicketStatus:"+LogCount+": Started");
LogCount = LogCount + 1;

// Find events in Netcool that have tickets which are not closed

Filter = "MaximoStatus < 4 and MaximoStatus > 1";
TicketEvent = GetByFilter("NCOMS_Alerts_Status",Filter,False);
NumEvents = length(TicketEvent);
if(NumEvents > 0){
  Count = 0;
  while( NumEvents > Count ){

    Filter = "TICKETID like "+TicketEvent[Count].MaximoTicketID+"";
    TicketStatus = GetByFilter("Maximo_Tickets", Filter, False);

    log("Maximo_UpdateTicketStatus:"+LogCount+ " Filter="+Filter);
    LogCount = LogCount + 1;

    // We will only get a result if the ticket status is different to the event
    NumTickets = length(TicketStatus);
    log("Maximo_UpdateTicketStatus:"+LogCount+ " Changed Status="+NumTickets);
    LogCount = LogCount + 1;

    if(NumTickets > 0){

      // Lookup conversion from severity value to name
      Results = GetByKey('SeverityConversion', TicketEvent[Count].Severity, 1);
    }
  }
}

```

```
// Check if Severity has changed and update Maximo
if(TicketStatus[0].OMNI_SEVERITY != Results[0].Severity){
    TicketStatus[0].OMNI_SEVERITY = ""+ Results[0].Severity;
};

// Check if the Maximo status has changed and update Netcool
if(TicketEvent[Count].MaximoTicketStatus != TicketStatus[0].STATUS){
    TicketEvent[Count].MaximoTicketStatus = ""+ TicketStatus[0].STATUS;
    TicketEvent [Count].MaximoTicketUID = ""+ TicketStatus[0].TICKETUID;

    if(TicketStatus[0].STATUS like 'CLOSED'){
        TicketEvent[Count].MaximoStatus = 4;
    };
};

} else {
// If none are changed simply exit without updating event
    Exit();
};

// Increase while count and check we are not in a loop
Count = Count + 1;
if(Count > 1000){
    Exit();
};

};
```

Troubleshooting

Use a database tool such as dbVisualiser to access the Maximo database and view the OMNIBUS2MAXIMO_IFA table. This should show a record for all of the processed tickets. New tickets will be shown in here as they are inserted by Impact and there will be a corresponding entry in MXIN_INTER_TRANS with the same TRANSID. As the crontask executes in Maximo it will read MXIN_INTER_TRANS and the entries will be removed. The records in OMNIBUS2MAXIMO_IFA however are not removed and this has to be done via Impact.

Once the message has been created by the crontask it is written to a queue for MEA to process. If for some reason invalid data is contained in the message MEA will not be able to process the information and it will cause the queue to grow. In this instance the messages should be deleted. This queue can be found in the webserver console at

Application Servers>MXServer>Messaging engines>ctgNode01.MXServer-intjmsbus>Queue points>CQINBD@ctgNode01.MXServer-intjmsbus>Messages