FS_Assets template

Template	Comments
{ "version": "2.0", "name": "ForeScout Assets Mgmt", "comment": "Assets Management", "type": "REST_EVENT", "event_type": ["LEASE", "FIXED_ADDRESS_IPV4", "HOST_ADDRESS_IPV4"], "transport": {"path": "/fsapi/niCore/Hosts"}, "action_type": "Assets Management", "content_type": "application/xml", "vendor_identifier": "ForeScout", "quoting": "XML",	“version” must be set to “2.0” (NIOS 8.1 supports version "2.0") This template can be used with LEASE, FIXED_ADDRESS_IPV4 and HOST_ADDRESS_IPV4 events/notifications. The API calls will use "/fsapi/niCore/Hosts" as a default path XML quoting is used by default.
"steps": [{ "name": "DebugOnStart", "operation": "NOP", "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}} ${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}} ${XC:DEBUG:{UT:}}" },	Steps block Debug output all variables in H, E, I, L, S, O, UT name spaces
{ "name": "assignSyncTime", "operation": "NOP", "body_list": ["${XC:COPY:{L:SyncDate}:{UT:TIME}}${XC:FORMAT:TRUNCATE:{L:SyncDate}:{16t}}"] },	Assign a local variable SyncDate which will be used to populate FS_SyncedAt extensible attribute

{
"name": "stop_if_just_changed",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${E:A:values{extattrs}{FS_SyncedAt}{value}}", "op": "==", "right":
"${L::SyncDate}"},
{"left": "${E:A:operation_type}", "op": "==", "right": "MODIFY"}
],
"stop": true}
},
	Stop the template execution if:
	- operation type is MODIFY
	- the object was recently synced

{
"name": "check_for_not_Lease",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
	Assign local variables if FIXED_ADDRESS_IPV4 or HOST_ADDRESS_IPV4 event is triggered

"statements": [
{"left": "${E::event_type}", "op": "!=", "right": "LEASE"},
{"left": "${E:A:values{extattrs}{FS_Sync}{value}}", "op": "==", "right":
"true"}
],
"eval":
"${XC:ASSIGN:{L:Sync}:{S:true}}${XC:COPY:{L:Site}:{E:values{extattrs}
{FS_Site}{value}}}${XC:COPY:{L:RemediateOnEvent}:{E:values{extattrs
}{FS_RemediateOnEvent}{value}}}${XC:COPY:{L:Obj_ref}:{E:values{_re
f}}}${XC:COPY:{L:IP}:{E:values{ipv4addr}}}${XC:COPY:{L:NV}:{E:values
{network_view}}}${XC:ASSIGN:{L:Obj_Ref_Add}:{S:}}",
"else_eval": "${XC:ASSIGN:{L:Sync}:{S:false}}"}
},
{
"name": "check_MAC",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${E::event_type}", "op": "!=", "right": "LEASE"},
{"left": "${E:A:values{extattrs}{FS_Sync}{value}}", "op": "==", "right":
"true"},
{"left": "${E:A:values{mac}}", "op": "!=", "right": ""}
],
"eval": "${XC:COPY:{L:MAC}:{E:values{mac}}}",
"else_eval": "${XC:ASSIGN:{L:MAC}:{S:000000000000}}"}
},	

{
"name": "check_for_Lease",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${E::event_type}", "op": "==", "right": "LEASE"},
{"left": "${E:A:ip.extattrs{FS_Sync}}", "op": "==", "right": "true"}
],
"eval":
"${XC:ASSIGN:{L:Sync}:{S:true}}${XC:COPY:{L:Site}:{E:ip.extattrs{FS_
Site}}}${XC:COPY:{L:RemediateOnEvent}:{E:ip.extattrs{FS_Remediate
OnEvent}}}${XC:COPY:{L:IP}:{E:values{address}}}${XC:COPY:{L:NV}:{
E:values{network_view}}}${XC:COPY:{L:MAC}:{E:values{hardware}}}"}
},
	Assign local variables if LEASE event is triggered

{
"name": "stop_if_no_sync",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [{"left": "${L::Sync}", "op": "==", "right": "false"}],
"stop": true}
},
	Stop template execution if synchronization is not required for an object

{
"name": "Goto for delete action",
"operation": "CONDITION",
"condition": {
"statements": [{"left": "${E:A:operation_type}", "op": "==", "right": "DELETE"}],
"condition_type": "AND",
"next": "DebugDelete"}
},
	If an object was deleted jump to "DebugDelete" step

{
"name": "check_for_Lease_go_for_Data",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [{"left": "${E::event_type}", "op": "!=", "right": "LEASE"}],
"next": "Get Discovery Data"}
},
	Jump to "Get Discovery Data" step if FIXED_ADDRESS_IPV4 or HOST_ADDRESS_IPV4 event is triggered

{ "name": "Create Obj_Ref for Lease", "operation": "SERIALIZE", "serializations": [{"destination": "L:Obj_Ref","content": "lease"}, {"destination": "L:Obj_Ref_Add","content": "&address=${L:A:IP}"}] },	Create variables for an API request to retrieve a discovery and user data for a lease. For other object these variables were previously defined
{ "name": "Get Discovery Data", "operation": "GET", "transport": {"path": "${L:A:Obj_ref}?_return_fields=discovered_data${L:A:Obj_Ref_Add}"}, "wapi": "v2.6" },	Request discovery data for an object
{ "name": "check_discoverer", "operation": "CONDITION", "condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{discoverer}}", "op": "!=", "right": ""}], "eval": "${XC:COPY:{L:discoverer}:{P:discovered_data{discoverer}}}", "else_eval": "${XC:ASSIGN:{L:discoverer}:{S:.}}"}}, { "name": "check_discovered_name", "operation": "CONDITION", "condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{discovered_name}}", "op": "!=", "right": ""}], "eval": "${XC:COPY:{L:discovered_name}:{P:discovered_data{discovered_name}}}", "else_eval": "${XC:ASSIGN:{L:discovered_name}:{S:.}}"}}, { "name": "check_ v_switch", "operation": "CONDITION", "condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{v_switch}}", "op": "!=", "right": ""}],	Populate local variables with discovered data. ForeScout CounterACT does not accept an empty parameter or parameter with a space, to simplify the template "." (a dot) is used as an empty value.

```
        "eval":
"${XC:COPY:{L:v_switch}:{P:discovered_data{v_switch}}}", "else_eval":
"${XC:ASSIGN:{L:v_switch}:{S:.}}"}},

{ "name": "check_ v_host", "operation": "CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{v_host}}", "op": "!=", "right": ""}],
        "eval": "${XC:COPY:{L:v_host}:{P:discovered_data{v_host}}}",
"else_eval": "${XC:ASSIGN:{L:v_host}:{S:.}}"}},
{ "name": "check_ v_datacenter", "operation": "CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{v_datacenter}}", "op": "!=", "right": ""}],
        "eval":
"${XC:COPY:{L:v_datacenter}:{P:discovered_data{v_datacenter}}}",
"else_eval": "${XC:ASSIGN:{L:v_datacenter}:{S:.}}"}},

{ "name": "check_ v_entity_name", "operation": "CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{v_entity_name}}", "op": "!=", "right": ""}],
        "eval":
"${XC:COPY:{L:v_entity_name}:{P:discovered_data{v_entity_name}}}",
"else_eval": "${XC:ASSIGN:{L:v_entity_name}:{S:.}}"}},
{ "name": "check_ v_adapter", "operation": "CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{v_adapter}}", "op": "!=", "right": ""}],
        "eval":
"${XC:COPY:{L:v_adapter}:{P:discovered_data{v_adapter}}}",
"else_eval": "${XC:ASSIGN:{L:v_adapter}:{S:.}}"}},
{ "name": "check_ v_entity_type", "operation": "CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{v_entity_type}}", "op": "!=", "right": ""}],
        "eval":
"${XC:COPY:{L:v_entity_type}:{P:discovered_data{v_entity_type}}}",
"else_eval": "${XC:ASSIGN:{L:v_entity_type}:{S:.}}"}},
{ "name": "check_ network_component_ip", "operation": "CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{network_component_ip}}", "op": "!=", "right": ""}],
        "eval":
"${XC:COPY:{L:network_component_ip}:{P:discovered_data{network_c
omponent_ip}}}", "else_eval":
"${XC:ASSIGN:{L:network_component_ip}:{S:.}}"}},
{ "name": "check_ network_component_name", "operation":
"CONDITION",
        "condition": {"condition_type": "AND", "statements": [{"left":
"${P::discovered_data{network_component_name}}", "op": "!=", "right":
""}],
        "eval":
"${XC:COPY:{L:network_component_name}:{P:discovered_data{networ
k_component_name}}}", "else_eval":
"${XC:ASSIGN:{L:network_component_name}:{S:.}}"}},
{ "name": "check_ network_component_port_name", "operation":
"CONDITION",
```

<table>
<tr>
<td>"condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{network_component_port_name}}", "op": "!=", "right": ""}],
"eval": "${XC:COPY:{L:network_component_port_name}:{P:discovered_data{network_component_port_name}}}", "else_eval": "${XC:ASSIGN:{L:network_component_port_name}:{S:.}}"}},
{ "name": "check_ network_component_port_description", "operation": "CONDITION",
"condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{network_component_port_description}}", "op": "!=", "right": ""}],
"eval": "${XC:COPY:{L:network_component_port_description}:{P:discovered_data{network_component_port_description}}}", "else_eval": "${XC:ASSIGN:{L:network_component_port_description}:{S:.}}"}},
{ "name": "check_ device_vendor", "operation": "CONDITION",
"condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{device_vendor}}", "op": "!=", "right": ""}],
"eval": "${XC:COPY:{L:device_vendor}:{P:discovered_data{device_vendor}}}", "else_eval": "${XC:ASSIGN:{L:device_vendor}:{S:.}}"}},
{ "name": "check_ device_model", "operation": "CONDITION",
"condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{device_model}}", "op": "!=", "right": ""}],
"eval": "${XC:COPY:{L:device_model}:{P:discovered_data{device_model}}}", "else_eval": "${XC:ASSIGN:{L:device_model}:{S:.}}"}},
{ "name": "check_ device_type", "operation": "CONDITION",
"condition": {"condition_type": "AND", "statements": [{"left": "${P::discovered_data{device_type}}", "op": "!=", "right": ""}],
"eval": "${XC:COPY:{L:device_type}:{P:discovered_data{device_type}}}", "else_eval": "${XC:ASSIGN:{L:device_type}:{S:.}}"}},
{
"name": "DebugDiscovery",
"operation": "NOP",
"body": "${XC:DEBUG:{P:}}${XC:DEBUG:{L:}}"
},</td>
<td></td>
</tr>
<tr>
<td>{
"name": "Get User Data",
"operation": "GET",
"transport": {"path": "networkuser?user_status=ACTIVE&address=${L:A:IP}"},
"wapi": "v2.6"
},
{ "name": "check_user_response", "operation": "CONDITION",
"condition": {"condition_type": "AND", "statements": [{"left": "${P:L:PARSE}", "op": "==", "right": "0"}],
"next": "check_username"}},
{</td>
<td></td>
</tr>
</table>

"name": "Pop User from the list",
"operation": "VARIABLEOP",
"variable_ops": [{
"operation": "UNSHIFT",
"type": "DICTIONARY",
"destination": "L:user",
"source": "P:PARSE"
}]},
{ "name": "check_username", "operation": "CONDITION",
"condition": {"condition_type": "AND", "statements": [{"left":
"${L::user{name}}", "op": "!=", "right": ""}],
"eval":
"${XC:COPY:{L:username}:{L:user{name}}}${XC:COPY:{L:domainname
}:{L:user{domainname}}}", "else_eval":
"${XC:ASSIGN:{L:username}:{S:.}}${XC:ASSIGN:{L:domainname}:{S:.}}
"}},
{
"name": "DebugUserData",
"operation": "NOP",
"body": "${XC:DEBUG:{P:}}${XC:DEBUG:{L:}}"
},	

{ "name": "Create_FS_Asset", "operation": "POST", "body_list": ["<?xml version=\"1.0\" encoding=\"UTF-8\"?>", "<FSAPI TYPE=\"request\" API_VERSION=\"1.0\">", "<TRANSACTION TYPE=\"update\">", "<OPTIONS CREATE_NEW_HOST=\"true\"/>", "<HOST_KEY NAME=\"ip\" VALUE=\"${L:A:IP}\"/>", "<PROPERTIES>", "<PROPERTY NAME=\"IB_MAC\"><VALUE>${L:A:MAC}</VALUE></PROPERTY>", "<PROPERTY NAME=\"IB_Comment\"><VALUE>Added via IB OutboundAPI at ${L:A:SyncDate}</VALUE></PROPERTY>", "<TABLE_PROPERTY NAME=\"IB_Location\"><ROW>", "<CPROPERTY NAME=\"Site\"><CVALUE>${L::Site}</CVALUE></CPROPERTY>", "<CPROPERTY NAME=\"Discoverer\"><CVALUE>${L::discoverer}</CVALUE></CPROP ERTY>", "<CPROPERTY NAME=\"Discovered name\"><CVALUE>${L::discovered_name}</CVALUE></CPROPERTY >", "<CPROPERTY NAME=\"Virtual Machine Name\"><CVALUE>${L::v_entity_name}</CVALUE></CPROPERTY>", "<CPROPERTY NAME=\"Virtual Datacenter\"><CVALUE>${L::v_datacenter}</CVALUE></CPROPERTY >", "<CPROPERTY NAME=\"Virtual Host\"><CVALUE>${L::v_host}</CVALUE></CPROPERTY>",	Send request to ForeScout and check the response

"<CPROPERTY NAME=\"Attached Device Address\"><CVALUE>${L::network_component_ip}</CVALUE></CPROPERTY>",
"<CPROPERTY NAME=\"Attached Device Name\"><CVALUE>${L::network_component_name}${L::v_switch}</CVALUE></CPROPERTY>",
"<CPROPERTY NAME=\"Attached Device Port\"><CVALUE>${L::v_adapter} ${L::network_component_port_name}</CVALUE></CPROPERTY>",
"<CPROPERTY NAME=\"Device Vendor\"><CVALUE>${L::device_vendor}</CVALUE></CPROPERTY>",
"<CPROPERTY NAME=\"Device Model\"><CVALUE>${L::device_model}</CVALUE></CPROPERTY>",
"<CPROPERTY NAME=\"Device Type\"><CVALUE>${L::device_type} ${L::v_entity_type}</CVALUE></CPROPERTY>",
"</ROW></TABLE_PROPERTY>",
"<TABLE_PROPERTY NAME=\"IB_User\"><ROW>",
"<CPROPERTY NAME=\"Username\"><CVALUE>${L::username}</CVALUE></CPROPERTY>",
"<CPROPERTY NAME=\"Domain\"><CVALUE>${L::domainname}</CVALUE></CPROPERTY>",
"</ROW></TABLE_PROPERTY>",
"</PROPERTIES>",
"</TRANSACTION>",
"</FSAPI>"
],
"parse": "XMLA"
},
{
"name": "Check add/modify",
"operation": "CONDITION",
"condition": {
"statements": [{ "left": "${P:A:PARSE{FSAPI}{STATUS}{CODE}}", "op": "!=", "right": "FSAPI_OK"}],"condition_type": "OR",
"error": true}
},	

{
"name": "stop_if_Lease",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [{"left": "${E::event_type}", "op": "==", "right": "LEASE"}],
"stop": true
}},
	Stop the template execution if the event type is Lease

{ "name": "next_if_Fixed", "operation": "CONDITION",	Looking _ref for parent object of HOST_ADDRESS_IPV4

"condition": {
"condition_type": "AND",
"statements": [{"left": "${E::event_type}", "op": "==", "right":
"FIXED_ADDRESS_IPV4"}],
"next": "Update Sync Time"
}},
{
"name": "Get HostIPv4 _ref",
"operation": "GET",
"transport": {"path":
"record:host?ipv4addr=${L:U:IP}&network_view=${L:U:NV}"},
"wapi": "v2.6"
},
{
"name": "Get_Objref",
"operation": "CONDITION",
"condition": {
"statements": [{"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}],
"condition_type": "AND",
"eval": "${XC:COPY:{L:Obj_ref}:{P:PARSE[0]{_ref}}}"}
},	

{ "name": "Update Sync Time", "operation": "PUT", "transport": {"path": "${L:A:Obj_ref}"}, "wapi": "v2.6", "wapi_quoting": "JSON", "body_list": ["{","\"extattrs+\":{\"FS_SyncedAt\": { \"value\": \"${L:A:SyncDate}\"}}","}"] },	Update FS_SyncedAt extensible attribute in NIOS
{ "name": "Stop Create/Modify", "operation": "CONDITION", "condition": {"statements": [{"left": "1", "op": "==", "right": "1"}],"condition_type": "AND","stop": true} },	Stop the template execution for INSERT operation

{
"name": "DebugDelete",
"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{
L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}"
},
	Debug variables for DELETE operation

{ "name": "Delete_FS_Asset", "operation": "POST", "body_list": ["<?xml version=\"1.0\" encoding=\"UTF-8\"?>", "<FSAPI TYPE=\"request\" API_VERSION=\"1.0\">",	Update IB_Delete property on ForeScout, which should trigger a policy on ForeScout and delete the object

"<TRANSACTION TYPE=\"update\">",
"<OPTIONS CREATE_NEW_HOST=\"true\"/>",
"<HOST_KEY NAME=\"ip\" VALUE=\"${L:A:IP}\"/>",
"<PROPERTIES>",
"<PROPERTY
NAME=\"IB_Delete\"><VALUE>Delete</VALUE></PROPERTY>",
"</PROPERTIES>",
"</TRANSACTION>",
"</FSAPI>"
],
"parse": "XMLA"
},
{
"name": "check delete",
"operation": "CONDITION",
"condition": {
"statements": [{ "left": "${P:A:PARSE{FSAPI}{STATUS}{CODE}}", "op":
"!=", "right": "FSAPI_OK"}],"condition_type": "OR",
"error": true}
}]
}	