NVS Greenex user guide

Edition 2023.12.21.11.39.15

Content

Introduction

General information

NVS Greenex Java platform

Introduction

NVS Greenex is a software platform written in Java and designed to solve a wide range of problems in different subject areas.

The modular structure allows you to combine different functions and use the maximum number of common basic functions such as

- Managing users and data access permissions
- Working with various types of databases
- Editing existing code and writing new programs.
- Dynamic compilation of programs
- A system for transferring changes across the landscape.

When developing it, have been taken into account the operating features and ease of use of such “big” technologies as SAP ™ and ORACLE ™

Thanks to Java technology, the system can run on different operating systems LINUX, Windows and use databases

- Postgresql > 14.5
- MySql 5.7(MariaDb 10.2.25)
- Oracle 12c
- SAP HANA > 2.04

However, this version of the system is limited to supporting only Linux and the Postgresql database. Other versions are in the testing stage.

Each system has a built-in performance test - PT01, with which you can evaluate the processing speed of multi-threaded tasks on a given server.

Use the built-in help system for a more detailed description of the procedure.

The platform actively uses sending data by mail and has built-in support for sending letters via SMTP and MAPI protocols.

For interaction between systems, the platform provides an API program interface, with which you can receive data or control some processes remotely.

For example, transferring a change request from the development system to the quality control system.

Access to API functions, as well as other transactions, is based on roles. You can grant both read and write permissions when working with database tables.

The built-in code editor SX38 is intended mainly for checking the correctness of the code in the system and making minor corrections. The development process involves working in an external IDE, Eclipse or similar.

With subsequent import of the jar file into the system. The process is described in more detail in the Developer's Guide and demo videos on YouTube.

The system is not demanding on resources, and can be installed even on 4 GB of RAM and 40 GB of disk space.

NVS Greenex - technical monitoring

Introduction

NVS Greenex - technical monitoring is a separate module designed to monitor the technical parameters of systems, mainly SAP and Oracle.

Such as

- Free disk space
- Loading RAM and CPU
- Availability of current backups
- Etc.

The basic set of the package contains the most necessary and most important check programs, but you can create your own by copying existing templates.

Historically, the NVS Greenex system was developed to monitor the health of SAP® and Oracle®. When developing, the operating experience of existing systems was taken into account

- SAP Solution manager
- Nagios
- Zabbix

The name refers to the green color of the indicators, similar to the dashboard of your car. All lights are green and your ride is smooth and fast.

When developing the monitor, the main efforts were aimed at increasing operational reliability while reducing the labor intensity of administration.

To achieve this goal, the developers chose a model of minimal use of intermediate agents that are required to receive data from the monitored systems.

Due to the exclusion of such agents, there is no need to install, configure and administer them, which significantly reduces the complexity of operating the monitoring system.

In this version of the monitor, installation of additional NSClient agents is required only for servers running the Windows® operating system.

In other cases, data is transmitted by request to the system via the ssh, https protocol or using the API of the server itself, which simplifies the interaction scheme, allowing for better stability in the operation of the entire monitoring scheme.

However, another important issue during development was taking into account the experience of the influence of the human factor on the process.

To minimize the risk of being ignored, in addition to standard operational warnings, the monitor provides the ability to schedule daily regular reports.

This report contains summary data with all currently unresolved problems.

The key feature of the format is there is no need to have technical knowledge about the essence of the problem, it is enough to react to the color of the report.

Like a car dashboard, the report should not have red. .

The report is sent by mail to several recipients, which reduces the risk of ignoring warnings from the monitoring system.

In addition, the daily report is a means of self-monitoring of the monitor itself, since it, in turn, is a system that itself may stop working for technical reasons.

Mutual monitoring of man and equipment can significantly reduce the risk of missing important information that warns of an increasing problem.

Eliminating the source of a problem in advance is much easier than correcting missed consequences.

User interface.

Introduction

The NVS Greenex platform has 3 ways of interacting with the user:

- Using a regular browser (Chrome, Firefox, Edge) via a secure protocol https
- Via a GUI graphical user interface created based on the Java Swing graphic library with data transfer via an internal protocol via sockets.
- Mobile application on Android that allows you to work with the system remotely via smartphone.

In the current version of the program, the first method is mainly used. The second method, as it is more labor-intensive and requires installation of a separate program on the user's computer, is considered as a backup.

Launch the program

Like any website on the Internet, the NVS Greenex interface is accessed using the URL bar in the browser, for example:

- https://192.168.10.125:8000/
- https://192.168.10.125:8000/sx01
- https://192.168.10.125:8000/sx01?item_id=5

where

192.168.10.125- IP address or hostname.

:8000 - port number. The last two digits are determined by the instance number when the system was installed.

sx01 - transaction name.

?item_id=5 - optional parameter(s).

A convenient way to launch is to create a .bat file in the Windows environment: The file name will determine the name of the transaction to be launched. Below is an example

```
set tcode=%~n0
set url=192.168.10.125:8000
"C:\Program Files\Mozilla Firefox\firefox.exe" -new-window "https://%url%/%
tcode%?login=ADMINISTRATOR&pwd=Zaqwerty13&item_id=7"
start chrome "https://%url%/%tcode%?login=ADMINISTRATOR&pwd=Zaqwerty13&item_id=7"
```

or use .KeePass

Transactions

To access various functions and data, the NVS Greenex system uses transactions - short character-numeric keys associated with specific parts of the program code. Consider them as global menu items.

The presence of a particular transaction is determined by the modules installed in the system and may differ in different installations. The set of basic functions defined in the program core remains relatively unchanged, such as

- SX01 user management,
- PXCG-roles and permissions,
- SX09, SXMS change management.

A complete list of basic functions can be found in the HELP help system.

To move from one transaction to another, use the window at the top left of the screen.

Passing parameters

As already mentioned, you can pass GET parameters in the call line for faster access to the desired data. The set of parameters depends on the specific transaction; however, most of them accept the item_id parameter.

For example, in the user editing transaction SX01 it will correspond to the user, in PXCG - maintaining roles it will call a specific role.

You can find out the required identifier by using the search 🔍 and writing it down from the line in the browser.

Security issues when passing parameters

Despite the fact that data exchange between the NVS Greenex system and the user is carried out using the secure https protocol and is transmitted over the network in encrypted form,

⚠use GET parameters (parameters in the call line) with login and password, for example: login=MYUSER&pwd=MyPassword123

recommended for test systems only

because passwords entered in this way can be visible in technical logs on the server.

For productive systems, enter the password at the initial "Welcome to NVS Greenex" invitation screen.

Monitoring

Equipment

SRV1 Servers

Introduction

With this transaction, you can manage a list of physical or virtual servers for the purpose of monitoring their status.

The NVS Greenex system distinguishes three types of objects:

- Servers (type S - servers).
- Databases (type D - db_systems).
- Application servers (type A app_systems).

Actions with servers

Use the  button on the left control panel to add a new server or  to search for an existing one.

Enter hostname as the description.

Checking connection

To monitor the status of each server, a user-password pair is required to communicate with it.

Enter the required parameters using the  button. Enter the password twice.

The Add info(optionally) field is not used here, leave it blank.

Save the data and check the connection results using the  button

If the communication details are entered correctly, you should see a response with a green flag on the console at the bottom right.

View monitoring results

Using the button you can see the results of background jobs dedicated to this server.

Current records, i.e. for which less time has passed than the task period, have green highlight.

If you need to clarify the details of the task, follow the link with the task ID.

Parameter: operating system types

To simplify things, the NVS Greenex-monitoring system does not strictly maintain a list of operating systems. However, it should adhere to the following rule, which is used by built-in check programs. The case of letters does not matter.

- All systems based on Microsoft Windows ® must contain the windows keyword
- All systems based on Linux ® must contain the linux keyword

Windows: monitoring agent

The NVS Greenex monitoring system is built on the principle that the simpler it is, the more reliable it is and therefore almost does not use intermediate agents.

However, if you are monitoring Windows, you will still need to install an intermediate agent NSClient++

During installation, select the web server type: in this case, monitoring will be able to read agent data via the https protocol.

Security question

To communicate with monitored systems, NVS Greenex uses secure communication protocols: https and ssh. All user passwords are stored in a separate table sys_passwords in encrypted form and are not suitable for use without a secret key on the server.

However, to reduce the risks of unauthorized connections, prevent attacks on equipment, and effectively monitor strictly it is recommended to use accounts with the minimum required permissions.

In each case, this will depend on the type of verification and will be achieved using the mechanisms of roles and permissions.

Tracing can help in achieving the goal, on the basis of which it is possible to create the necessary roles for the monitor.

DBS1 Databases

Introduction

With this transaction, you can manage a list of database servers for health monitoring purposes.

The NVS Greenex system distinguishes three types of objects:

- Servers (type S - servers).
- Databases (type D - db_systems).
- Application servers (type A app_systems).

Database actions

Use the  button on the left control panel to add a new server database or  to search for an existing one.

As a description, you can enter, for example, sid + hostname.

Checking connection

Monitoring the status of each database requires a user-password pair to communicate with it.

Enter the required parameters using the  button. Enter the password twice.

The Add info(optionally) field is not used here, leave it blank.

Save the data and check the connection results using the  button If the communication details are entered correctly, you should see a response with a green flag  on the console at the bottom right.

View monitoring results

Using the  button you can see the results of background jobs dedicated to this database.

Current records, i.e. for which less time has passed than the task period, have green highlight. If you need to clarify the details of the task, follow the link with the task ID.

Parameter:database type

To simplify things, the NVS Greenex-monitoring system does not strictly maintain a list of database types. However, it should adhere to the following rule, which is used by built-in check programs. The case of letters does not matter.

- All systems based on SAP HANA ® must contain the keywords sap and hana. Port is 3[SysNr]13.
- All systems based on Oracle ® must contain the oracle keyword
- All systems based on Postgres ® must contain the postgres keyword
- All systems based on MySql(MariaDb) ® must contain the mysql keyword (mariadb)
- All systems based on Microsoft Sql® must contain the keywords microsoft and sql

Security issues

To communicate with monitored systems, NVS Greenex uses secure communication protocols: https and ssh.

All user passwords are stored in a separate table sys_passwords in encrypted form and are not suitable for use without a secret key on the server.

However, to reduce the risks of unauthorized connections, prevent attacks on equipment, and effectively monitor strictly

It is recommended to use accounts with the minimum required permissions. In each case, this will depend on the type of verification and will be achieved using the mechanisms of roles and permissions.

Tracing can help in achieving the goal, on the basis of which it is possible to create the necessary roles for the monitor.

APP1 Application servers

Introduction

With this transaction, you can manage a list of application servers for health monitoring purposes.

The NVS Greenex system distinguishes three types of objects:

- Servers (type S - servers).
- Databases (type D - db_systems).
- Application servers (type A app_systems).

Actions with application servers

Use the  button on the left control panel to add a new server application or  to search for an existing one.

You can enter, for example, sid + hostname as a description, but the choice is yours.

Checking connection

To monitor the status of each application server, a user-password pair is required for communication.

Enter the required parameters using the  button. Enter the password twice.

The Add info(optionally) field is used for SAP ® systems; in the case of an ABAP connection type, add additional parameter: client, for example: CLNT=001.

Save the data and check the connection results using the  button

If the communication details are entered correctly, you should see a response with a green flag  on the console at the bottom right.

View monitoring results

Using the  button you can see the results of background jobs dedicated to this database.

Current records, i.e. for which less time has passed than the task period, have green highlight.

If you need to clarify the details of the task, follow the link with the task ID.

Parameter:application type</h3>

To simplify things, the NVS Greenex-monitoring system does not strictly maintain a list of application types. However, it should adhere to the following rule, which is used by built-in check programs. The case of letters does not matter.

- All systems based on SAP ABAP ® must contain the keywords sap and abap
- All systems based on Oracle ® must contain the oracle keyword

Security issues

To communicate with monitored systems, NVS Greenex uses secure communication protocols: https and ssh. All user passwords are stored in a separate table sys_passwords in encrypted form and are not suitable for use without a secret key on the server.

However, to reduce the risks of unauthorized connections, prevent attacks on equipment, and effectively monitor strictly it is recommended to use accounts with the minimum required permissions.

In each case, this will depend on the type of verification and will be achieved using the mechanisms of roles and permissions.

Tracing can help in achieving the goal, on the basis of which it is possible to create the necessary roles for the monitor.

MPRJ Project management.

With this transaction you can add, delete and edit projects for equipment lists.

Project management becomes relevant when it is necessary to logically separate groups of equipment or information systems to be able to send alerts to personnel responsible for a particular function.

In transaction AS12, the mentioned projects can be linked to each email address, for which notifications will be sent.

Tasks (background jobs)

BJ01 regular background jobs

Introduction

You can schedule a recurring task in the system using the built-in task scheduler.

Each button has a tooltip when you hover over it.

Creating a new task

To create a new background job

- Click on the add  button at the top left.
- Enter a short description of the task, such as "check Linux disks."
- The system will create an initial entry that appears on the left. After this you will be able to:
- Set the program name, start time, execution period. To do this, select the desired option on the right
- and edit its value using the context menu or the  button.

Search for an existing task

To search for an existing task, use the  button

Start and stop

To run a task, change its status (task_status) to Active (ready to schedule):

To stop - Inactive (suspended).

View log

The task execution log can be seen using the last results button 

If the job is active and running, then the last, topmost entry has green background , which suggests that less than the task period has passed since the recording.

If you need to find records for a specific object, use the context menu item show results for object

Sending email notifications

In some cases, to prevent unwanted spam, sending email notifications (alerts) can be blocked  or unblocked  in general for the entire task.

List of task objects

Each task can work with its own list of objects. For example, a disk scan job will work with the list servers, and the task of checking current backups with its own list of databases. To create such lists, the transaction BJ02 is intended

List of task parameters

Any parameters used in the program code can be specified for the task. To create such lists, the transaction BJ03 is intended

Mail mailing recipients

In the case of sending email notifications while a task is running, the list of recipients can be specified using transaction AS12

BJ02 monitoring objects

Introduction

Each task usually has its own list of objects (servers, databases, applications) with which it works

Some of the jobs work with only one type, such as servers, while others are complex and can include different types.

Working with objects

To include a server in the task list

- having selected the task on the left, click the list of servers:  or through the context menu add server
- add a new server  , or delete from the list 
- For similar actions with the database, use 
- For similar actions with application servers, use 

BJ03 additional job parameters

Introduction

With this transaction you can specify additional parameters for the background job The names of the parameters and their values are determined by the functionality of the program that this task runs.

Working with parameters

Using the buttons on the control panel at the top left you can

- Create new 
- Delete unnecessary 
- Copy all existing parameters to a new job 

Description of parameters

Parameter names and their usage are determined by the report code.

Program	Parameter name	Description	Example
report.monitoring.nvs.com.RegularReport1	fs_jobs	File system check IDs	fs_jobs=1,3,5
report.monitoring.nvs.com.RegularReport1	backup_jobs	IDs for checking backups	backup_jobs=2

Types of checks

Base Package.

Check program list

Checks for Linux

Program name	Description
checksLinux.monitoring.nvs.com.CheckLinuxDisks	Free space on Linux disks (file systems)
checksLinux.monitoring.nvs.com.CheckLinuxCpu	Linux CPU usage percentage
checksLinux.monitoring.nvs.com.CheckLinuxRam	Linux RAM usage percentage

Checks for Windows

Program name	Description
checksWindows.monitoring.nvs.com.CheckWindowsDisks	Free space on Windows disks.

Checks for Oracle database

Program name	Description
checksOracle.monitoring.nvs.com.CheckOracleBackup	Number of hours since last backup
checksOracle.monitoring.nvs.com.CheckOracleSize	Database size
checksOracle.monitoring.nvs.com.CheckOracleTablespace	Availability of free space in Oracle tablespaces

Checks for SAP HANA database

Program name	Description

checksSapHana.monitoring.nvs.com.CheckSapHanaBackup	Number of hours since last backup
checksSapHana.monitoring.nvs.com.CheckSapHanaReplicationStatus	Replication status
checksSapHana.monitoring.nvs.com.CheckSapHanaSize	Database size

Checks for SAP ABAP Application Server

Program name	Description
checksSapAbap.monitoring.nvs.com.CheckSapAbapSm13	Update records in SM13
checksSapAbap.monitoring.nvs.com.CheckSapAbapSm50	Percentage of busy processes in SM50
checksSapAbap.monitoring.nvs.com.CheckSapAbapSt22	Dynamics of short dumps in ST22

Common types.

Program name	Description
checksOther.monitoring.nvs.com.CheckHostPing	Check ping
checksOther.monitoring.nvs.com.CheckHPServiceGuardStatus	Check HP Serviceguard cluster status
checksOther.monitoring.nvs.com.CheckPostgresSize	Checking Postgres database size
checksOther.monitoring.nvs.com.CheckWebPageIsAlive	Checking html page availability

Linux disks

checksLinux.monitoring.nvs.com.CheckLinuxDisks

Monitor get information by ssh : "df -k"

You can set system parameter maxUsedPercentDisksLimit to change default value 85% or specify separate limit in transaction LM10 .

Linux CPU

checksLinux.monitoring.nvs.com.CheckLinuxCpu

Monitor gets information by ssh : "top -b -n 1 | head -4"

You can set system parameter maxCpuUsedPercentLimit to change default value 95%.

Linux RAM

checksLinux.monitoring.nvs.com.CheckLinuxRam

Monitor get information by ssh : "cat /proc/meminfo"

Windows disks

checksWindows.monitoring.nvs.com.CheckWindowsDisks

⚠ In order to use this type of check you need installing additional third-party product NSCP++

You can check connection to windows server with NSCP++ client by means of curl

For example

```
curl -k -i -H "Accept: application/json" -H "Password:Init1234" -X GET
"https://192.168.10.151:8443/query/check_drivesize"
```

current version is NSCP-0.5.2.35

The most important settings in C:\Program Files\NSClient++\nsclient.ini are:

```
[/settings/default]

password = Init1234
allowed hosts = *
[/settings/WEB/server]

port = 8443

[/modules]

CheckDisk = enabled
```

You can set system parameter maxUsedPercentDisksLimit to change default value 85% or specify separate limit in transaction LM10 .

Oracle backup

checksOracle.monitoring.nvs.com.CheckOracleBackup

Monitor gets information from SQL :

```
select s1.* from (select round((sysdate - end_time) * 24) as output  from v$rman_status
where operation = 'BACKUP' and object_type in('DB FULL')  and status='COMPLETED' order by
start_time desc) s1
where rownum=1
```

You can set system parameter BackupMaxPastHoursAllowedLimit to change default value 48 hours or specify separate limit in transaction LM10 ..

For access to view in Oracle add follow permissions:

grant select on SYS.V_$RMAN_STATUS to <monitoring_user_name>;

Oracle db size

checksOracle.monitoring.nvs.com.CheckOracleSize

Monitor gets information from SQL :

```
select round((sum(bytes)/1048576/1024),2) output from v$datafile
```

You can specify separate limit in transaction LM10 which will send alert when datafile volume exceeds it.

For access to view in Oracle add follow permissions:

grant select on SYS.V_$DATAFILE to <monitoring_user_name>;

Oracle tablespace

checksOracle.monitoring.nvs.com.CheckOracleTablespace

Monitor gets information from SQL

```
select
substr(df.TABLESPACE_NAME,1,15) as TABLESPACE,
substr(df.FILE_NAME,1,50) as DATA_FILE,
df.AUTOEXTENSIBLE as EXT,
TO_CHAR((df.maxbytes/1024/1024),'99999990') as MAX_MB,
TO_CHAR((df.BYTES/1024/1024),'99999990') as SIZE_MB,
TO_CHAR((s1.free_size/1024/1024),'99999990') as FREE_MB
from dba_data_files df
left join
(select fs.FILE_ID,
sum(fs.BYTES) as free_size
from dba_free_space fs
group by fs.FILE_ID) s1
on s1.FILE_ID = df.FILE_ID
order by df.TABLESPACE_NAME,df.FILE_ID
```

You can specify separate limit on single tablespace in transaction LM10 .

Also you can set system parameter maxUsedPercentOracleTablespace in sx21 which sets limit for all tablespaces .By default it is 90%.

For access to view in Oracle add follow permissions

```
grant select on dba_data_files to &lt;monitoring_user_name&gt;;
grant select on dba_free_space to &lt;monitoring_user_name&gt;;
```

How to test

Add test tablespace to database:

```
CREATE TABLESPACE tbs1  DATAFILE 'C:\oracle\oradata\PRD\tbs1_data.dbf' SIZE 1m AUTOEXTEND
OFF;
```

Create table:

```
CREATE TABLE t1(id INT GENERATED ALWAYS AS IDENTITY, c1 VARCHAR2(32)) TABLESPACE tbs1;
```

Fill table with data:

```
 BEGIN
  FOR counter IN 1..10000 loop
   INSERT INTO t1(c1)
   VALUES(sys_guid());
  END loop;
 END;
/
```

You will recieve alert from monitor.

Next add datafile to tablespace:

```
ALTER TABLESPACE tbs1 ADD DATAFILE 'C:\oracle\oradata\PRD\tbs1_data2.dbf' SIZE 1m
AUTOEXTEND OFF;
```

or

switch on autoextend on the first datafile

```
ALTER DATABASE  DATAFILE 'C:\oracle\oradata\PRD\tbs1_data.dbf' AUTOEXTEND ON MAXSIZE 2m ;
```

Monitor will send recovery message.

SAP HANA backup

checksSapHana.monitoring.nvs.com.CheckSapHanaBackup

Monitor gets information from SQL :

```
select SECONDS_BETWEEN (SYS_END_TIME, NOW()) / 3600 as past_hours
from "SYS"."M_BACKUP_CATALOG"
where entry_type_name in ('complete data backup','incremental data backup')
and state_name='successful' order by sys_start_time desc LIMIT 1;
```

You can specify separate limit in transaction LM10 which will send alert when datafile volume exceeds it.

For access to data in SAP HANA add follow permissions:

```
CREATE USER &lt;monitoring_user_name&gt; PASSWORD &lt;pass&gt; NO
FORCE_FIRST_PASSWORD_CHANGE;
ALTER  USER &lt;monitoring_user_name&gt; DISABLE PASSWORD LIFETIME;
GRANT MONITORING TO &lt;monitoring_user_name&gt;;
```

SAP HANA replication

checksSapHana.monitoring.nvs.com.CheckSapHanaReplicationStatus

Monitor gets information from SQL

```
SELECT CASE WHEN (select count(*) from SYS.M_SERVICE_REPLICATION)= (select count(*) from
SYS.M_SERVICE_REPLICATION
WHERE SECONDARY_FULLY_RECOVERABLE='TRUE')  THEN 1 ELSE 1000 END AS output FROM dummy
```

In case the replication is interrupted monitor receives 1000 as result and sends alert.

For access to data in SAP HANA add follow permissions

```
CREATE USER &lt;monitoring_user_name&gt; PASSWORD &lt;pass&gt; NO
FORCE_FIRST_PASSWORD_CHANGE;
ALTER  USER &lt;monitoring_user_name&gt; DISABLE PASSWORD LIFETIME;
GRANT MONITORING TO &lt;monitoring_user_name&gt;;
```

SAP HANA db size

checksSapHana.monitoring.nvs.com.CheckSapHanaSize

Monitor gets information from SQL :

```
 select host,usage_type,round(used_size/1024/1024/1024) "size_gb"  from M_DISK_USAGE
where usage_type='DATA'
```

You can specify separate limit in transaction LM10 which will send alert when datafile volume exceeds it.

For access to data in SAP HANA add follow permissions

```
CREATE USER &lt;monitoring_user_name&gt; PASSWORD &lt;pass&gt; NO
FORCE_FIRST_PASSWORD_CHANGE;
```

```
ALTER  USER &lt;monitoring_user_name&gt; DISABLE PASSWORD LIFETIME;
```

```
GRANT MONITORING TO &lt;monitoring_user_name&gt;;
```

SAP ABAP SM13

checksSapHana.monitoring.nvs.com.CheckSapAbapSm13

Monitor counts records in ABAP table VBHDR :

The system parameter maxSm13RecordsLimitSapAbap = 100 by default. If there are more records than limit you'll get alert. You can change it in SP21.

You can specify separate limit in transaction LM10 which will send alert when datafile volume exceeds it.

How to check

Import test program <Install Folder>/utils/SAP_test_reports.zip

Go to se38 and start Z_NEMONITOR01 with T_PAUSE=300 sec. Next, go to SM13 and sure that there are update records. Monitor has to record total count in log.

SAP ABAP SM50

checksSapHana.monitoring.nvs.com.CheckSapAbapSm50

Monitor gets information from SAP module TH_WPINFO :

The system parameter maxBusyPercentLimitSapAbap = 75 by default. If there are more records than limit you'll get alert. You can change it in SP21.

You can specify separate limit in transaction LM10 which will send alert when datafile volume exceeds it.

How to check

Import test program <Install Folder>/utils/SAP_test_reports.zip

Go to se38 and start Z_NEMONITOR02 with T_PAUSE=300 sec. Next, go to SM50 and make sure that there are many busy work processes. Monitor has to send an alert.

SAP ABAP ST22

checksSapHana.monitoring.nvs.com.CheckSapAbapSt22

Monitor reads records from table SNAP :

The system parameter maxSt22RecordsLimitSapAbap = 30 and pastMinitesSm22SapAbap = 15 by default.

If there are more records than limit you'll get alert. You can change it in SP21.

You can specify separate limit in transaction LM10 which will send alert when datafile volume exceeds it.

How to check

Import test program <Install Folder>/utils/SAP_test_reports.zip

Go to se38 and start Z_NEMONITOR02 with T_NUMS > maxSt22RecordsLimitSapAbap and second report Z_NEMONITOR04 which will devide by 0.

Next, go to SM22 and make sure that there are many short dumps. Monitor has to send an alert.

Ping

checksOther.monitoring.nvs.com.CheckHostPing

the program checks the availability of the socket (connectivity)

HP SG status

checksOther.monitoring.nvs.com.CheckHPServiceGuardStatus

the program checks the status of HP Service Guard cluster packages via ssh

Postgresql db size

checksOther.monitoring.nvs.com.CheckPostgresSize

the program checks the size of the database

Html page status 200

checksOther.monitoring.nvs.com.CheckWebPageIsAlive

the program checks the status of the html page (200 OK)

LM10 maintaining limits

Introduction

By default, all checks use the general parameters specified in SP21 or the instance profile.

With this transaction, you can make more fine-grained settings by setting separate limits for individual checks and objects.

Create a limit,  assign the following parameters to it

- jobId - check number in BJ01
- descr - description (optional parameter)
- objectId - Object ID
- objectType - object type: server S, database D or application server A.
- limit_key - limit code. Text value.
- limit_value - limit value. Digital value.

For example

- 1 - Id for checking Linux file systems BJ01
- volume of the database log directory
- 185 - Id of the server with the mentioned database
- S
- /disk2/log
- 20

As a result, the program will not use the general criterion of file system fullness of 85%, but will generate an alert when the 20% limit is reached. The meaning of limit processing depends on the algorithm of the verification subroutine and requires clarification in each specific case.

Regular reports

Report 1.

Introduction

This report is used to regularly distribute monitoring results to assess the current state of the monitored systems.

Using it, you can choose which types of checks to include in the report and when and how often to send to interested parties letters with a visual presentation of the results in the form of a table.

Creating a report

In transaction BJ01 create a new task  , in the program_name field specify report.monitoring.nvs.com.RegularReport1

using the context menu, go to transaction BJ03 and add the parameter fs_jobs or backup_jobs in which you list the ID's of the checks.

For example,

fs_jobs=1,2,3 - the report will include the results of the latest file system checks from jobs 1-3.

backup_jobs=5 - the report will contain checks of backups from task 5.

In transaction BJ01 edit the start time, period in seconds, status-A active. (scheduled_at, period_sec,task_status).

In transaction AS12 create the required email if necessary and add the report (task) created in the first step from the context menu.

If necessary, you can check the correctness of sending emails through the send test email item.

After this, at the specified time and with the specified frequency, you should receive regular letters with the report.

Report 2.

Introduction

This report is used to regularly distribute monitoring results to assess the current state of the monitored systems.

Using it, you can choose which types of checks to include in the report and when and how often to send to interested parties letters with a visual presentation of the results in the form of a table.

Creating a report

In transaction BJ01 create a new task  , in the program_name field specify report.monitoring.nvs.com.RegularReport2

using the context menu, go to parameters. Next again via context menu select add another job and type key word from job description which results you want to recieve by this report.

In AS12 select necessary email and via the context menu, add job that you have created.

After this, at the specified time and with the specified frequency, you should receive regular letters with the report.

Alerts

ALR1 view alerts.

Introduction

With this transaction you can view and mark all alerts in the system as processed. In the future, all processed alerts will be deleted automatically.

Incidents in SAP Solman

SSM1 integration with Solman.

Introduction

With this transaction you can create incidents using a simplified procedure.

Clicking on the link from the report allows you to automatically fill in data about the problem and send a request to create an incident for support in SAP Solution manager.

Permissions are checked on the SAP side, so you must have the required role for this. If necessary, contact to the SAP Solution Manager administrator.

In this case, each application is stored in a local table. For security reasons, the SAP password is used only for sending applications via the API and is not saved in the monitor database.

Creating a request in SAP SM

If this transaction is called with a result_id parameter that points to a record in the monitor table mon_single_result, then at startup it opens partially completed form with data on the problem.

Fill in the user name and password in SAP Solution Manager and click the SEND button.

If the operation is successful, you will see a message in the form of a green flag and the number of the created incident in Solution Manager. 

If there is an error, such as authorization, additional information can be found in the sx21 system log or in SAP (ST22,SM21)

View created requirements for an application in the monitor

Using the  button or through the context menu you can view the last 20 records stored in the monitor table.

View created orders in SAP SM

To find the created ticket in SAP Solution Manager,

run transaction CRM_UI in SAP GUI,

on the page that opens in the browser on the left Managing change requests >> change documents.

In the search criteria ID. For the document, indicate the ID received from SAP, or enter the criterion Period = today.

The created application should appear in the search results.

Settings

To integrate with SAP, you must have the following parameters

Parameter name in sp21	>Example	Description
SapSolmanSSM1	SOLMAN_PRD	System name in the sys_remote_systems table
SapSolmanMonUser	PUBLIC	Public user name for logging in to monitoring
SapSolmanMonPwd	Zaqwerty12	public user password for logging in to monitoring

Prerequisites

In order to create incidents in SAP Solution Manager you have to import TR in <Install Folder>/utils/SAP_SM_create_incident.zip which contains function Z_SM_CREATE_INCIDENT.

Email notifications

Setting up mail sending.

Introduction

NVS Greenex can send letters via a mail server using the SMTP and MAPI Microsoft Exchange protocols.

The sendMailTransport parameter in the starting instance profile is responsible for the sending method.

If you are using SMTP

Add the parameter sendMailTransport=JavaMail to the instance profile start_profile.properties.

Add the following parameters to transactions SP21

Parameter name	Example
mail.smtp.host	smtp.google.com
mail.smtp.port	25
mail.smtp.auth	true
mail.smtp.starttls.enable	true
mail.smtp.user	info@nvs-itech.com
mail.smtp.password	***
mail.smtp.from	info@nvs-itech.com

If you are using MS Exchange

Add the parameter sendMailTransport=MsExchange to the instance profile start_profile.properties.

Add the following parameters to transactions SP21

Parameter name	Example
MsExchangeWebLink	https://email.yourcompany.com/ews/Exchange.asmx

MsExchangeUser	postman
MsExchangePassword	***
MsExchangeDomain	yourdomain

If none of the options are set, mail is saved to the SX21 system log.

f you have questions related to sending mail, consult your mail service providers.

AS12 address maintenance.

Introduction

With this transaction you can manage the list of recipients who will receive notifications generated by system. In an emergency, you can disable/enable notifications in the background

task, managing the disable_email status using buttons  or  in transaction

Actions with addresses

Use the  button on the left control panel to add a new address

Select an unnecessary address in the list on the right and use the  button to delete.

Use the toggle context menu, change the status to disable/enable the ability to send letters to this address.

Use the add job context menu, add a background job that will send letters to this address.

Use the add project context menu, add a project to receive letters to this address about events with its objects.

Use the add all context menu, add the ability to receive all letters sent by the system.

Basic functions

SX38 Java Editor

NVS Greenex uses open source Java code which is stored in a database.

With this transaction you can create, edit and compile existing routines or create your own to extend the functionality of the system.

Development in the NVS Greenex environment is a separate big topic.

You can find more details in the Developer Guide.

SX01 User management

Introduction

With this transaction, you can create, edit system users, and assign roles to them to separate access rights.

- Create a new user using 
- Edit an existing 
- Or delete unnecessary 

Using the context menu, add the desired role that you created in the PXCG transaction.

All permissions to access system objects are determined by a set of roles.

When logging in for the first time, the user will be required to change the password to his own.

PXCG Maintaining roles and permissions

With this transaction, you can create, edit user roles, and assign permission objects to them to separate access rights.

 Create role

 Delete role

 Edit role

 Show a list of all roles in the system.

Use the context menu to add a new permission object to a role.

SX09 Creating transport requests

With this transaction, you can create transport requests, include system objects in it for transport across the landscape.

NVS Greenex supports 2 ways to transfer changes across the landscape

- development package DM01,SP09 - for consistent transfer of several objects at the same time
- single change request SX09 - for single objects (project, role).

Create a new change request 

Include any object in it, for example a project via a context menu item.

After this, release the request 

As a result, a .zip file will appear with the objects included in it in the transport directory, ready for further transfer across the landscape.

Copy the file to the transport directory of the second system and import the data through transaction SXAM.

SXMS Importing transport requests

As well as importing packages in the SXAM transaction, you can use this SXMS transaction to import single change requests.

When launched, the transaction will check the contents of the transport directory /usr/nvs/trans/in

If necessary, set the correct permissions for the files in the specified directory (change the owner to [sid]adm)

Use the context menu to

- Import single request
- Only view the contents of the request without importing

SXAM Installing service packs

With this transaction, you can install additional update modules to expand the functionality of the system.

When opened, the transaction reads the list of available packages in the transport transaction /usr/nvs/trans/sxam

Using the context menu you can

- import package contents into system,
- or just view its contents without importing.

Use the  button to view already imported packages.

SX04 User sessions

Introduction

With this transaction you can control the activity of user sessions in the system.

The system records the user's login time, logout time, and what transactions he used.

When working through a browser, there is no concept of a permanent session, so every 15 seconds the user's browser sends the IamAlive signal, by which the system determines that the session is active.

If the timeout is exceeded and there is no response, the system considers the session terminated.

To correctly end the session, use the  button.

SX12 Managing locks

Some transactions use locks when working with system objects to prevent parallel users from changing the object at the same time.

In a normal situation, when the user session is correctly terminated by pressing the  button, the locks should be deleted automatically.

If for some reason this does not happen, you can manually remove them from the lock table using this transaction.

SX20 Login log.

With this transaction, you can see the user's name, login and logout times, and the transactions he started in the system log.

By default, audit data is stored for 30 days.

SX21 System log.

Globally, the Greenex NVS system records all error messages and additional information in its main log: <install_dir>/log/logger.trc.

For example: /usr/nvs/DEV_D00/log/logger.trc

You can access it using either transaction SX21 or browsing through the operating system.

SX53 Authorization check.

In the NVS Greenex system, there is access distribution based on a role matrix.

If the user does not have enough permissions for certain actions, she writes the result to the file <install_dir>/tmp/<user_name>/<user_name>.sx53

for example/usr/nvs/DEV_D00/tmp/ADMINISTRATOR/ADMINISTRATOR.sx53

Use this transaction to check the result of the authority check if the program does not return the expected data.

SX05 Program trace.

To debug a program, in some cases you may need detailed additional information that tracing can provide.

If you want to enable tracing, create a file <install_dir>/tmp/<user_name>/<user_name>.trc

For example

```
su - devadm -c "touch /usr/nvs/DEV_D00/tmp/ADMINISTRATOR/ADMINISTRATOR.trc"
```

In the file you can see the texts of the SQL queries being executed, the results of permission checks and much other useful information for debugging the program.

⚠️Don't forget to delete the file after use, otherwise its uncontrolled growth may lead to disk space problems

PT01 Performance test.

Introduction

With this transaction, you can evaluate system performance by measuring the duration of a standard test using the built-in indicator.

Testing procedure

Select the Settings item at the top left and use the clear result table context menu to clear the previous results, if any.

Run calculate test 1 and calculate test 2 in turn to evaluate performance.

Change the parameters in the table at the top right if necessary to evaluate the performance margin.

- parallel_threads - number of parallel processes.
- calculate_value - number of degrees to calculate radians. Used to set high CPU load.
- repeat_cycles - the number of repetitions of calculations in each process.
- jobs_number - total number of test jobs.

Evaluation of results

As a result of the test, the calculated values of the coefficient and its checksum in the results table must coincide. If there is an error, the likely cause will be a discrepancy between the database and/or operating system settings and the specified values in the test. See the SX21 system log for more information. The test records the results in a single spt_calculation table and does not affect other data in any way.

WARNING: Do not run the test while the system is productive. It will not damage existing data, however, with some initial values will be able to "hang" system up to the need for a complete reboot

SP21 System parameters.

Introduction

With this transaction you can add, delete, edit parameters stored in the sys_parameters table of the database.

These settings work alongside the settings from the instance's startup profile and require a system reboot for them to take effect.

-  Add new parameter
-  Delete parameter

Via the context menu you can change the value or set it in encrypted form.

SP09 Creation of distribution and support packages.

Introduction

With this transaction, you can export program code and data from the development system to distribution packages.

Steps to create a distribution

The list of objects to be included in packages is contained in the sys_modules and sys_modules_components tables

SQL query for reconciliation: You need to make sure that all the tables needed to be included in the package are in the table. The is_special field means that the package will additionally include data from the table.

```
select a.table_name,b.object_name FROM information_schema.tables a
left join sys_modules_components b on a.table_name = b.object_name and
b.object_type='table'
where a.table_schema = 'public';
```

SX11: Reimport all records using the command in transaction SX11 -> reimport all DDIC records for tables. This synchronizes the data dictionary with the actual database structure.

SP21: Check that the parameters sp09_basis_export_folder for the base distribution and sp09_spam_export_folder for creating support packages are correct. Directories for uploading data must exist and have write permissions.

SP09: Select the required package on the left and execute Release module. Check the result in the upload directories. In case of errors, additional information can be found in the SX21 system log or <install_folder>/log/logger.trc

Development

Developer"s guide.

Introduction

The platform NVS Greenex is builded on Open JDK Java 1.8 (Server part) with using JavaScript,HTML,CSS as user interface via browser.

System contains build-in code editor (SE38) where you can create simple programs but its main goal is to correct small program errors.

In order to make you job more convenient and comfortable it's recommended to organize development workplace on your PC.

Take in account that the first setup can take a while but you'll save a lot of time in the future avoiding repeated steps.

You will be need some additional programs for it

- Eclipse IDE for Java. This is the main tool for development. It has rich possibilities for rapid and effective creating your programs (Java classes). If you are comfortable with another IDE it is up to you.
- Apache Ant. Very helpfull utility to compile and build your programs into jar (Java archive) files.
- Open JDK 1.8. Needs for Ant. You are free to use any distribution.
- Heidi SQL client. for working with Postgres database. You can use any one like this.
- Putty and plink. Tools for connect to Linux and transfer your jar files on development system.
- Any browser : Firefox, Chrome or Edge. In order to avoid many times logons it's recomended create .bat file. If you OS is not MS Windows, all steps will be the same or a little bit differ.

Prerequisites

Current version of NVS Greenex requires using Postgres Database as a development system. Some development functions won't work on others DBs.

As test and production system you can choose: Postgres,MariaDb (MySQL),Oracle,SAP Hana - you are able transfer data on your landscape with transport requests and support packages.

There is also an alternative technology of User Inteface based on Java Swing library, but it has no any profits and is more complex in development and support.

Developer workplace preparation

Install Eclipse IDE for Java Developer

Install Apache Ant and set environment variables

Install OpenJDK 1.8

Install Heidi SQL client

Install Putty

Install your own development system or use existing one in your company.

Creating the first program

Here we'll create new user program Test01.

Create forlder where you'll save all your developments:

Copy program template and unzip it into C:\dev

Edit program name to Test01

Correct path in ant/build.xml to Open JDK

Copy 2 common libraries NvsBasis.jar and NvsShared.jar from DEV system into C:\dev_lib and make a link.

Edit .bat file with correct parameters DEV system and mount points. Here you'll be need having any network shared disk or setup samba on DEV system. It's up to you.

Open Test01 in Eclipse:

Edit project build path to add link to the common libraries:

Correct package and classes names according new name Test01. Rename package with your company name. Don't use nvs.com, please!

Change program caption to be sure your minimal program works.

Add record to table sys_web_links. You can ommit guid of project, we add it later.

Run .bat file. It will compile (ant.exe) your program into jar file (folder out) and copy program on shared disk, then connect via plink.exe on Linux, put jar into lib/ folder and restart service.

Save your own template to use it in the future. Next time it will require less afforts. Just use copy/paste method.

First run

Try to start your program in browser

You can add new authorization object into already existing role or create a new one. Let's create role

Add new role to user

Now we are able to open our new program Test01 in browser:

Import program into database

NVS Greenex supports two ways storing its code: in .jar file in /lib or in tables sys_jc_projects/sys_jc_classes/sys_jc_resources of database. You can use any way but the second one is recommended.

If program loads from jar file, it has dark blue caption, when from database - turquoise.

In case the program (class) exists in the database and in lib/*.jar then the last will be loaded.

Let's presume we've fixed all bugs in our new program Test01 and are going to transfer code on landscape: Create new project in SE38.

Import jar file into project. Compile. Take into account that jar file will be deleted from lib/. After import you have to restart system. Use restartService.sh script for it.Add project guid to table sys_web_links. It is necessary for further transfer all data into transport request. There is also command from menu in SE38 to create link for project.

Program propagating. Transfer on the landscape

Create transport request in SE09 and include in it project Test01 and TEST_ROLE.

Release request. New zip file will be created in /usr/nvs/trans/out/

Transfer request into TEST(QAS) system. Just copy zip file to /usr/nvs/trans/in/ and change linux permissions to 777.

Import request via STMS.

Add role TEST_ROLE to user and run Test01.

Creating your own support packages

NVS Greenex supports three ways to propagate its programs (objects).

- You can just copy your programs in .jar files into /lib every system. System restart is required.
- You can create projects in SE38 and include all objects into transport request (SE09->STMS). System restart is not required.
- You can create own support package and transfer data via SP09->SPAM. It's recommended for large development. System restart is not required.

Let's imagine you have new one development which includes several tables with contents, roles and projects.

To be sure that we transfer fresh code , let's change caption of Test01 in SE38

Next, create a table in DEV system via SQL client:

Import table structure into data dictionary (sys_dic_columns). You can do it for one table or all tables simultaneously:

Create records for two packages SP00 and SP01. First will be content table(s) with initial data, second - all objects wich you'll change many times in the future:

Release your packages in SP09, edit parameters if you need:

Copy zip files /usr/nvs/trans/spam/ in test system, change permissions, look at content in SPAM:

Import your packages one by one, make sure there are new table and new version of Test01 in TEST system:

Workspace agreement

In order to avoid any conflicts of names in system in the future, use, please

"YourCompany.com" package name in Java code. Don't use nvs.com.

Use Z or Y as the first simbols in

transaction (sys_web_links), for example use "ZTEST01" or "YTEST01" instead of "TEST01".

name of role (sys_roles), for example use "ZROLE1" or "YROLE1" instead of "ROLE1".

name of tables (sys_dic_tables). for example use "ZTABLE1" or "YTABLE1" instead of "TABLE1".

TCode template project

To start development your own application in NVS Greenex with Eclipse you can use a template from <Install Folder>/utils/tcode_example.zip

Copmpilaton script examples.

Here are examples of .bat files on Windows for developers in order to make process easier.

To use it, you'll need to installPutty

With these scripts you can automate manually actions from transaction SP09,SXMS.

Export development package into file

File destination will be determined with variable sp09_spam_export_folder (tcode SP21).In example this is w:\temp\packages

```
set packageName=APIDEV-01
set devSysIp=192.168.10.145
set rootPwd=Zaqwerty12

set "cmd=/usr/nvs/DEV_D00/jvm/bin/java -jar /usr/nvs/DEV_D00/TransCmd.jar export_package
%packageName%"
"C:\Program Files\PuTTY\plink.exe" -batch -ssh root@%devSysIp% -pw %rootPwd% "%cmd%"
```

Import development package into another system

```
:: copy package on remote server and import

set packageName=APIDEV-01
set qasSysIp=192.168.10.150
set rootPwd=Zaqwerty12
set osUser=qasadm
set sysName=QAS_D05
set sysPath=/usr/nvs/%sysName%



:: Copy package .zip to transport directory

"C:\Program Files\PuTTY\pscp.exe" -pw %rootPwd% -P 22 -sftp
w:\temp\packages\%packageName%.zip root@%qasSysIp%:/usr/nvs/trans/spam/



:: Set permissions
set "cmd=chown -R %osUser%:greenex /usr/nvs/trans/"
"C:\Program Files\PuTTY\plink.exe" -batch -ssh root@%qasSysIp% -pw %rootPwd% %cmd%
```

```
:: Import into destination system
set "cmd=su - %osUser% -c '%sysPath%/jvm/bin/java -jar %sysPath%/TransCmd.jar
import_package %packageName%.zip'"
"C:\Program Files\PuTTY\plink.exe" -batch -ssh root@%qasSysIp% -pw %rootPwd% %cmd%
```

Transfer jar files between systems

```
:: transfer library MyLibrary.jar

set jarFileName=MyLibrary.jar
set qasSysIp=192.168.10.150
set rootPwd=Zaqwerty12
set osUser=qasadm
set sysName=QAS_D05
:: Copy from temporary folder into  destination folder in QAS ./lib/
"C:\Program Files\PuTTY\pscp.exe" -pw %rootPwd% -P 22 -sftp w:\temp\jars\%jarFileName%
root@%qasSysIp%:/usr/nvs/%sysName%/lib/

:: Set permissions
set "cmd=chown -R %osUser%:greenex /usr/nvs/%sysName%"
"C:\Program Files\PuTTY\plink.exe" -batch -ssh root@%qasSysIp% -pw %rootPwd% %cmd%

:: Restart service
set "cmd=systemctl restart greenex_%sysName%.service"
"C:\Program Files\PuTTY\plink.exe" -batch -ssh root@%qasSysIp% -pw %rootPwd% %cmd%
```

API

API Application Programming Interface.

Introduction

The current version of the system supports API via the https protocol. You can call a function using a standard GET request by passing in the line input parameters and receiving a response in JSON or PLAIN TEXT format.

List of functions:

Function name	Description	Input parameters
get_sysinfo	Returns system information	cmd=get_sysinfo

Call examples

Example call from Java:

```
https://ADMINISTRATOR:Zaqwerty13@192.168.10.131:8000/api?cmd=api.basis.nvs.com.GetSysInfo
Handler&#38;param1=3567&#38;param2=90
```

Incoming API requests are processed by the pages.nvsBasis.nvs.com.ApiHandler class. It checks the user's authority to call the function and loads the function handler class passed as the cmd parameter.

List of functions

Function name	Parameters	Description
api.basis.nvs.com.GetSysInfoHandler		System information

api.basis.nvs.com.RemoteTRImportHandler	tr_guid - guid of file,tr_name - request name	transport request import

Transport system

SX59 remote systems.

The system has a list of remote systems with which communication is carried out, as well as login information (login/password)

Use this transaction to register such systems and organize them into groups.

- Create new 
- Show list 
- Delete a system from the list 

To check the connection, you can use one of the items in the context menu.

TR10 transport routes.

Introduction

This transaction allows you to create a list of systems included in the transport landscape

Possible 2- or 3-link landscape for transfer

- DEV -> PRD
- DEV -> QAS -> PRD

Terrain creation

Select Transport route on the left and use the context menu to create one of two landscapes:

In the form that opens, select DEV, QAS and PRD systems. If the systems you need are not available,

add them to transactions sx59. You will then be able to transfer transport requests directly from the current transaction.

Transport Request Transfer

Along with using the usual method (SXMS transaction or TransCmd.jar), you can transfer queries remotely, similar to SAP CTS

Select the DEV system and use the context menu to transfer the desired request to QAS or PRD.

⚠This version uses file transfer via the socket mechanism, so the function can only work within a network (intranet).

TransCmd transport utility.

Introduction

The system includes the transport utility TransCmd.jar, which allows you to transfer changes across the landscape by exporting/importing data using the command line.

You can use it as an alternative to transactions SP09,SX09,SXMS,SXAM to create automated scripts.

Run it without parameters to get quick help

```
./jvm/bin/java -jar TransCmd.jar
```

Function name	Description	Example	Note
export_package	Exports a package from the system as a .zip file	/usr/nvs/DEV_D00/jvm/bin/java -jar TransCmd.jar export_package PACK-00	the export path is determined by the system variable sp09_spam_export_folder, PACK-00 package name
import_package	Imports data from .zip into the system	/usr/nvs/DEV_D00/jvm/bin/java -jar TransCmd.jar import_package PACK-00	.zip must be in the transport directory /usr/nvs/ trans/spam/
import_jar	Imports a development .jar file into the project and then compiles it	/usr/nvs/DEV_D00/jvm/bin/java -jar TransCmd.jar import_jar MoniChecksSP1.jar e7d2a8c2-fcd7-4e12-b5fc-7f0fe6c59d8f	.jar should located in the directory /usr/nvs/DEV_D00/lib/, MoniChecksSP1.jar - development file, e7d2a8c2-fcd7-4e12-b5fc-7f0fe6c59d8f - project guid
import_tr	Imports a transport request	/usr/nvs/DEV_D00/jvm/bin/java -jar TransCmd.jar import_tr QAS000001	the file must be in the directory /usr/nvs/trans/in/, QAS000001.zip - request file.

Thanks for your attention!