



uniPaaS Server for IBM i V1.9i Release Notes



Magic Software Enterprises Ltd provides the information in this document as is and without any warranties, including merchantability and fitness for a particular purpose. In no event will Magic Software Enterprises Ltd be liable for any loss of profit, business, use, or data or for indirect, special, incidental or consequential damages of any kind whether based in contract, negligence, or other tort. Magic Software Enterprises Ltd. may make changes to this document and the product information at any time without notice and without obligation to update the materials contained in this document.

Magic is a trademark of Magic Software Enterprises Ltd.

Copyright © Magic Software Enterprises, July 2012



Dear Developer,

We are pleased to present you with the release of uniPaaS Server V1.9i for the IBM i platform.

Please read the information below to learn more about uniPaaS Server for IBM i V1.9i.



Please note the PTF requirements for IBM i 6.1.

For information about uniPaaS V1.9i, see Magic Software Enterprise's [DevNet](#).



General Information.....	4
Prerequisites	4
Java Requirements	5
Installation	6
Installation Components	7
Library	7
IFS Directory	7
CGI Directory in QOpenSys.....	7
Configuration.....	8
Subsystem Configuration	8
Setting the uniPaaS Configuration File (MAGIC400.INI)	11
FQDN (Fully Qualified Domain Name).....	11
HTTP Server (Powered by Apache) Configuration	12
uniPaaS Requester Settings	12
Memory Allocation	13
Java Integration.....	14
Systinet Installation	14
WebSphere MQ	15
National Special Characters	16
Unicode	16
Using Colors in Browser Tasks	16
Browser Client	16
Server Initialization.....	17
Server Log File	18
New Features	19
Licensing.....	21
Performance – Managing Workload.....	21
Recommendations.....	21
Fixed QCRs.....	22
Limitations.....	22



General Information

uniPaaS for IBM i Server V1.9i lets you deploy uniPaaS applications directly on the IBM i server.

Prerequisites

1. OS/400 V5R4 ,OS i 6.1, or OS i 7.1
2. OS/400 Portable App Solutions Environment (PASE) – IBM License program

- 5722SS1 option 33 V5R4
- 5761SS1 option 33 6.1

PTF SI39074 and PTF SI43215 are required for OS I 6.1 (Product 5761SS1). The PTFs can be installed as individual fixes or as part of CUMULATIVE PTF PACKAGE C1256610 for IBM i 6.1.

- 5771SS1 option 33 7.1
3. IBM Java License program (see Java Requirements)
 4. TCP/IP configuration for IBM i
 5. TCP/IP Interface must be Active
 6. Local Host name must be defined in the host table entries
 7. Domain Name must be configured
 8. Domain Name must be added to host table entries
 9. Easycom Host Library
 - OS i V5R4 and 6.1: Host 1.1a or later
 - OS i 7.1 : Host 1.3

uniPaaS 1.9i has been tested with Host Library 1.3.

10. The User profile that will be used by the MGSERVER and MGBROKER jobs must have *RWX rights for most of the IFS directories in the uniPaaS 1.9 installation. After the installation, it can be set using the following commands:

```
CHGAUT OBJ('/UNIPAAS19') USER(USERPRF) DTAAUT(*RWX) OBJAUT(*ALL)  
SUBTREE(*ALL).
```

where USERPRF is a user name or user group profile that will be used for running the server jobs and UNIPAAS19 is the server directory.



11. When using the printing functionality of uniPaaS server for IBM i, make sure that the CCSID property of the user profile that runs the engine is not set to 65535. No output will be produced if you work with that CCSID.
12. When using graphical fields (Unicode) in your application, make sure that the user profile of the MGCLIENT job is not set to 65535.
13. Permanent or Evaluation License PAK

Java Requirements

If you use Java-based functionality in your application, it is necessary to start uniPaaS engines with the Java Virtual Machine. The **STRUNISVR** command has a new parameter, **Load JVM**. If you set this parameter to ***Yes**, Java prerequisites are mandatory. When **Load JVM** is set to ***No**, Java Requirements are not relevant. Java license programs are required to be installed on your IBM i server only when the uniPaaS application is designed to utilize Java-based functionality. For example:

- Java functions
- Providing Web services
- Web service consuming
- Using JMS – Java Messaging Server
- Using EJB – Enterprise Java Bean

Below are the Java requirements according to the operating system level:

For V5R4

1. IBM Developer Kit for Java License program 5722JV1 Option *Base
2. J2SE 6.0 32 bit – License program 5722JV1 Option 11
3. Java Group PTF SF99291 level 18 or later

For 6.1

1. IBM Developer Kit for Java License program 5761JV1 Option *Base
2. J2SE 6.0 32 bit – License program 5761JV1 Option 11
3. Java PTF Group SF99562 level 12 or later

For 7.1

1. IBM Developer Kit for Java License program 5761JV1 *Base Option
2. J2SE 6.0 32 bit – License program 5761JV1 Option 11
3. Java PTF Group SF99572 level 3 or later



Installation

1. Execute the **unipaas1.9i_iseriess.zip** file in your PC's **temp** directory.
2. Extract the **UNI019000** file from the zip archive.
3. Copy the **UNI019000** file into any IFS directory of your IBM i server using one of the methods below:
 - IBM i Client Access
 - IFS Windows network sharing
 - FTP
 - NFS Share
4. Install the product using the **INSMSEPRD** IBM i command.
 - a. If you previously installed the **MSEGPL** library with the **INSMSEPRD** command, type **MSEGPL/INSMSEPRD** from the IBM i command line and press **Enter**.
 - b. If you have not previously installed the **MSEGPL** library with the **INSMSEPRD** command, first follow the steps provided in the [Install MSE Product \(INSMSEPRD\) Tool](#) topic and then run the **MSEGPL/INSMSEPRD** command.
5. The Install MSE product display appears. At the Stream file prompt, specify the name of the STMF you downloaded in step 4.
6. Fill in the **Installation Name** parameter, which is the name of the new Library and IFS directory, for example: UNIPAAS19.
7. Fill in the root of the server's IFS directory. The default is **/**, which means that the server directory will be created in the IFS root, for example: /uniPaaS19.
8. Specify whether you need a CGI directory to be created in **/QOpenSys**. If you are planning to only use uniPaaS as a partitioning engine, then there is no need to create a directory in **QOpenSys**.
9. Fill in the **CCSID – ANSI** parameter, which is the CCSID of your environments, for example: 1252 for West European countries or 1250 for East European countries.
10. Sign off of your 5250 in order to clean QTEMP after the installation.

It is also possible to install the uniPaaS server without the INSMSEPRD command.

- a. Copy **UNI019000** to the IBM i save file (SAVF) using the File Transfer Protocol (FTP) in binary mode.
- b. Restore the content of the save file into the **QTEMP** library.
- c. **RSTOBJ OBJ(INSTALL*) SAVLIB(PRODUCT) DEV(*SAVF) OBJTYPE(*ALL) SAVF(QTEMP/PART9999) MBROPT(*ALL) ALWOBJDIF(*ALL) RSTLIB(QTEMP)**
- d. Run **QTEMP/INSTALL** and continue from step 6 above.



Installation Components

Library

- uniPaaS Server subsystem, Job Queues, Job Class, Job Descriptions, Message files, data areas
- CLLE interfaces to uniPaaS PASE environment, exit programs, service programs
- uniPaaS Server configuration menu (CFGMGCSVR) and relevant commands
- MGPRINT and MGPRINTC – Printing routine that converts IFS text-based files to spools

IFS Directory

- uniPaaS Server (bin/unirte)
- uniPaaS Broker (broker/u MGPRINT nirqbroker)
- uniPaaS command line requester (broker/mgrqcmdl)
- uniPaaS ISAM DB2/400 gateway (bin/mgeac/mgeacdbcs)
- uniPaaS SQL DB2/400 gateway (bin/mgdb400)
- uniPaaS memory gateway (bin/mgmemory)
- uniPaaS WS MQ gateway (bin/mgmqseries_c)
- uniPaaS CGI requester (cgibin/mgrqcgi019)
- uniPaaS UDF/UDP examples (userproc directory)
- uniPaaS Web utility files used for Browser Client support (web_utils directory)
- uniPaaS Hangul support (language/mglocal.kor)
- uniPaaS SNMP sub-agent (snmp/mgsnmp.so)
- uniPaaS Messaging component (messaging/messaging.ecf)
- Systinet server for Java (web_services directory)

CGI Directory in QOpenSys

- uniPaaS CGI requester (/QOpenSys/.../cgibin/mgrqcgi019)

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).



Configuration

You configure and manage the uniPaaS server using the Server administration menu.

Enter the following commands in the IBM i command line to invoke the uniPaaS Server administration menu:

1. Type **CHGCURLIB <New Library name>** and press **ENTER**.
2. Type **CFGMGCSVR** and press **ENTER**.

Subsystem Configuration

1. Configure the uniPaaS server using **Magic Config/Magic Subsystem Config**. At this stage, **JOBQ MGSERVER** and a routing entry for the **MGSERVER** job are added to a selected subsystem description. **MGSERVER JOBQ** is a default **JOBQ** of the server jobs. When you start a server job using **STRUNISVR**, **STRMAGRUN** or **STRMAGINST**, the job is submitted to **JOBQ MGSERVER**. The routing entry sets a job class and a memory pool of the system main storage where the **MGSERVER** job is going to run. By default, it is set to **MGSERVER** job class and storage pool **1**.

The following job properties are controlled by the job class: Run priority, Time slice in milliseconds, Eligible for purge, Default wait time in seconds, Maximum CPU time in milliseconds, Maximum temporary storage in megabytes, and Maximum threads. You can manually change them during the job performance tuning by using the **CHGCLS** IBM i command. For example: **CHGCLS CLS(UNIPAAS19/MGSERVER) RUNPTY(XX) TIMESLICE(XXXX) CPUTIME(*NOMAX) MAXTMPSTG(*NOMAX) MAXTHD(*NOMAX)**.

The routing entry is linked with a job description via the **Compare value** property of the routing entry and the **Routing Data** property of the job description. When the **MGSERVER** job enters the subsystem via **JOBQ**, the system validates the **Routing Data** of the job description against the **Compare value** of the routing entries in the subsystem. When the match is found, the job's memory pool is assigned and the relevant job properties are applied to the job according to the routing entry settings.

Make sure that the server subsystem is not active when you configure it via the menu. For security reasons, this step is allowed only when the subsystem is not active. **Make sure that you do not have jobs pending in the job queue.**

As an alternative to the configuration menu, you can use the IBM i **ADDJOBQE** or **RMVJOBQE** and **ADDRTGE** or **RMVRTGE** commands for uniPaaS server configuration:



- a. Connect JOBQ to subsystem:
**ADDJOBQE SBSL(SBSLIB/SBSNAME) JOBQ(UNIPAAS19/MGSERVER) MAXACT(5)
SEQNBR(10)**

Where:

SBSLIB – name of the subsystem library

SBSNAME – subsystem name

UNIPAAS19 – name of the server installation

5 – maximum number of concurrently active jobs

10 – job sequence number in subsystem property. The next available number can be found using the WRKSBSD command:

WRKSBSD SBSLIB/SBSNAME

option 5 (Display)

option 6 (Display Job Queue Entries)

- b. Disconnect JOBQ from Subsystem:
RMVJOBQE SBSL(SBSLIB/SBSNAME) JOBQ(UNIPAAS19/MGSERVER)
- c. Add Routing entry:
**ADDRTGE SBSL(SVRLIB/SBSNAME) SEQNBR(10) CMPVAL(UNI19LB.MGRNT)
PGM(QCMD) CLS(SVRLIB/MGSERVER)**
- d. Remove Routing entry:
RMVRTGE SBSL(SVRLIB/SBSNAME) SEQNBR(10)

2. If there's an error, verify the **joblog** using the **DSPJOB option 10**.

3. Configure the uniPaaS broker using **Magic Config/Broker Subsystem Config**.

At this stage, **JOBQ MGBROKER** and a routing entry for the **MGBROKER** job are added to a selected subsystem description.

MGBROKER JOBQ is a default job for the uniPaaS broker jobs. When you start a broker job using **STRMRBRUN**, the job is submitted to **JOBQ MGBROKER**. The routing entry sets a job class and a memory pool of the system main storage where the **MGBROKER** job is going to run. By default, it is set for the **MGBROKER** job class and storage pool **1**.

The following job properties are controlled by the job class: Run priority, Time slice in milliseconds, Eligible for purge, Default wait time in seconds, Maximum CPU time in milliseconds, Maximum temporary storage in megabytes, and Maximum threads. You can manually change them during the job performance tuning by using the **CHGCLS IBM i** command.

For example: **CHGCLS CLS(UNIPAAS19/MGBROKER) RUNPTY(XX) TIMESLICE(XXXX)
CPUTIME(*NOMAX) MAXTMPSTG(*NOMAX) MAXTHD(*NOMAX)**.



The routing entry is linked with a job description via the **Compare value** property of the routing entry and the **Routing data** property of the job description. When the **MGBROKER** job enters the subsystem via **JOBQ**, the operating system validates the **Routing Data** of the job description against the **Compare value** of the routing entries in the subsystem. When the match is found, the job's memory pool is assigned and the relevant job properties are applied to the job according to the routing entry settings.

Make sure that the subsystem is not active when you configure it via the menu.

For security reasons, this step is allowed only when the subsystem is not active.

Make sure that you do not have jobs pending in the job queue.

As an alternative to the configuration menu, you can use the IBM i **ADDJOBQE** or **RMVJOBQE** and **ADDRTGE** or **RMVRTGE** commands for uniPaaS broker configuration.

- a. Connect JOBQ to subsystem:

```
ADDJOBQE SBSLIB(SBSLIB/SBSNAME) JOBQ(UNIPAAS19/MGBROKER) MAXACT(1)  
SEQNBR(20)
```

Where:

SBSLIB – name of the subsystem library

SBSNAME – subsystem name

UNIPAAS19 – name of the uniPaaS 1.9i installation

1 – maximum number of concurrently active jobs

20 – job sequence number in subsystem property. The next available number can be found using the **WRKSBSD** command:

```
WRKSBSD SBSLIB/SBSNAME
```

option 5 (Display)

option 6 (Display Job Queue Entries)

- b. Disconnect JOBQ from the Subsystem:

```
RMVJOBQE SBSLIB(SBSLIB/SBSNAME) JOBQ(UNIPAAS19/MGBROKER)
```

- c. Add a Routing entry:

```
ADDRTGE SBSLIB(SVRLIB/SBSNAME) SEQNBR(20) CMPVAL(UNI19LB.MGRNT)  
PGM(QCMD) CLS(SVRLIB/MGBROKER)
```

- d. Remove a Routing entry:

```
RMVRTGE SBSLIB(SVRLIB/SBSNAME) SEQNBR(10)
```

3. If there's an error, verify the **joblog** using the **DSPJOB option 10**.



Setting the uniPaaS Configuration File (MAGIC400.INI)

1. The MAGIC400.INI file can be edited from the IBM i command line or from any other text editor.
2. Do not copy any special characters. This may cause abnormal behavior in the product.
3. You can copy the database definition in the **MAGIC_DATABASES** section from the MAGIC.INI file on Windows to the MAGIC400.INI file on IBM i.
4. It is highly recommended to backup the MAGIC.INI file before editing.
5. It is recommended to set the **READONLY** attribute of the MAGIC400.INI file. It can be done via the **QSH** or **PASE** command. For example: QSH CMD('chmod 444 /UNIPAAS19i/etc/MAGIC400.INI'). Make sure that the user profile that runs the server does not have ***ALLOBJ** rights.

FQDN (Fully Qualified Domain Name)

The broker and enterprise server should bind using a specific network adapter by specifying a FQDN (instead of an IP address). The requester layer should translate the FQDN to IP and bind using an IP on a specific adapter.

FQDN stands for fully qualified domain name – for example "Power6.mse.corp".

The MGREQ.INI file contains the following entry: **BindFirstIPAddress=Y/[N]**.

- Y – During binding to a port, the server will resolve the host name and will bind to the resolved IP address.
- N – The server will bind to any IP address (*.port – for backwards compatibility).

To enable a uniPaaS engine and broker to work with a specific network adapter (if there are multiple adapters on a machine):

1. Edit the MGREQ.INI file and enable **BindFirstIPAddress** (= Y) and set **MessagingServer** to **FQDN/port**.
2. Edit the MGRB.INI file and set **MessagingServer** to **FQDN/port**.
3. In the MAGIC.INI file, set **TCP/IP = 2, 30, 1500-2000 /LocalHost=FQDN**.
4. In the MAGIC.INI file, set the **Default Broker** to **FQDN/port**.



The table below shows the binding for the server module:

Port Number	V9.4sp6c-1 and above
Port-No	BindFirstIPAddress=N */Port-No BindFirstIPAddress=Y IP-address/Port-No
Ip Address/Port-No	IP-Address/Port-No

HTTP Server (Powered by Apache) Configuration

1. Start the administrative server instance by issuing the following command:
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
2. Open the following URL to create the Apache instance:
http://<server_name>:2001
3. The Apache instance directory and its configuration files are created in the **/www** directory. Copy the lines contained in **UNIPaaS19/web_utils/magic.conf** to the Apache configuration file.

Note: The MGREQ.INI file, which is used by Apache, is located in the same directory where the uniPaaS Internet requester program **mgrqcgi019** is located. The default location of the MGREQ.INI file and the uniPaaS Internet requester program **mgrqcgi019** is the **cgibin** directory.

uniPaaS Requester Settings

The following settings in the MGREQ.INI file affect the requester execution:

RetryMainTime=nn (default: 5)

The **RetryMainTime** setting determines the time interval that a requester connected to an alternate Request Broker should try to reconnect to the main Request Broker. If the requester is already connected to the main Request Broker, the **RetryMainTime** setting does not apply. The time interval is specified in minutes.

For example, when **RetryMainTime=15**, the requester tries to reconnect to the main Request Broker every 15 minutes. When the main Request Broker becomes available, the requester connects to it and disconnects from the alternate Request Broker. If the main Request Broker becomes unavailable, the requester returns to the alternate Request Broker and tries to reconnect to the Main Request Broker every 15 minutes.

When **RetryMainTime=0**, the requester does not try to connect to the main Request Broker, but remains connected to the alternate Request Broker.

KeepAlive=Y\N (default: N)



When **KeepAlive=Y**, the uniPaaS Generic Messaging Layer module lets the system administrator use operating-system level settings to control the keep-alive intervals for each open connection. Only a system administrator should determine the settings for the operating system intervals.

Memory Allocation

In the IBM i PASE, the number of segments that a process can use for data is controlled by the **LDR_CNTRL** environment variable. By default it is set to 256MB.

The default might not be sufficient for large uniPaaS applications (big ECF files) that use a memory database, parse big XML documents and so on.

In certain conditions, the job might need to exceed the limit. Regardless of the setting of the IBM i **MGSERVER** job description where we use the ***NOMAX** value for the temporary storage size, the PASE process terminates while exceeding the number of allowed memory segments to be used. In this case, we often observe an abnormal termination of the uniPaaS engine with the MCH6801 Object domain, a storage protection error, or a similar error.

To change the default, add one of the following entries into the **etc\mgenv400** file:

- **LDR_CNTRL=MAXDATA=0x10000000** – sets the limit to 256 MB
- **LDR_CNTRL=MAXDATA=0x20000000** – sets the limit to 512 MB
- **LDR_CNTRL=MAXDATA=0x30000000** – sets the limit to 768 MB
- **LDR_CNTRL=MAXDATA=0x40000000** – sets the limit to 1 GB
- **LDR_CNTRL=MAXDATA=0x50000000** – sets the limit to 1.24 GB
- **LDR_CNTRL=MAXDATA=0x60000000** – sets the limit to 1.5 GB
- **LDR_CNTRL=MAXDATA=0x70000000** – sets the limit to 1.75 GB
- **LDR_CNTRL=MAXDATA=0x80000000** – sets the limit to 2 GB



Java Integration

The Java CLASSPATH separator character on IBM i platforms is a colon (:) as opposed to the Windows platform separator character, which is a semicolon (;).

For example: CLASSPATH = /java/MyClasses:/java/OtherClasses

For more information, please refer to the [Java documentation](#).

For V5R4 only:

Make sure that you have installed Java Developer Kit 1.6.

To set Java 1.6 as your default Java:

- Specify **-Djava.version=1.6** on the Java command or in your **SystemDefault.properties** file; otherwise the Java will default to 1.4.

For V5R4, 6.1 and 7.1:

You can run the 32-bit Java 1.6 by specifying the **JAVA_HOME** environment variable.

Systinet Installation

1. Log on to the OS400 target machine.
2. Run the command:

```
ADDENVVAR ENVVAR(JAVA_HOME)  
VALUE('/QOpenSys/QIBM/ProdData/JavaVM/jdk60/32bit')
```

3. Run the command: CALL PGM(QP2TERM).
4. Go to "<magic installed directory>\web_services.
5. Edit the ssjinstallconfig file as following:

- Change the "dname" parameter from "dname=<MachineName>.<DomainName>" to real machine and domain names such as "dname=POWER6"
- Change the "installation.destination" parameter from "<SSJ Installation Directory>" to real SSJ installation directory such as "installation.destination=/UNIPAAS19I/ssj"

6. Run the command:

```
java -jar wso2-soa-enablement-server-java-6.6.jar -u ssjinstallconfig
```



7. There is no need to copy any files to the ssj directory. Just start SSJ from the command line:

```
CHGCURLIB SrvLib
```

(where SrvLib = uniPaaS server library).

```
SrvLib/STRSTNET SBS(SbsLib/SbsName) SSJ('/ssjdir')  
JAVAHOME('/QOpenSys/QIBM/ProdData/JavaVM/jdk60/32bit/jre')
```

where

SbsLib = Subsystem Library
SbsName = Subsystem Name
ssjdir = SSJ directory

8. Validate active SSJ using the following URL: <http://System:6060>
9. To end SSJ, run:

```
SrvLib/ENDSTNET SBS(SbsLib/SbsName) SSJ('ssjdir')  
JAVAHOME('/QOpenSys/QIBM/ProdData/JavaVM/jdk60/32bit/jre')
```

WebSphere MQ

uniPaaS 1.9i has been created and tested based on WMQ V6. On the IBM i platform, the WebSphere MQ connection can be established in the server mode only. IBM does not provide a “native” WebSphere MQ client.

If you are using a WebSphere MQ Queue manager on IBM i and you have to configure uniPaaS for communication with MQ, you need to set a logical name in the MAGIC400.INI file:

WMQ_ModuleName = C.

To connect to the WebSphere MQ Queue manager on another IBM i system or on any other platform, you have to interface with WebSphere MQ classes for Java using uniPaaS’s Java integration features.



National Special Characters

The uniPaaS installation is configured by default to generate output in the West European character set: ISO-8859-1, which is equivalent to ANSI 1252. If you have a requirement to support a different character set (such as ISO-8859-2 or ISO-8859-3), please contact our Support department to get assistance in configuring the following:

- MAGIC400.INI
- AS2EB
- ACS400
- Codepage.dat
- IBM i user profile and Job CCSID

Unicode

While working with Unicode Blobs in uniPaaS, you need to take into account the differences when handling Blob fields in UTF16 encoding.

- On a Windows-based system, UTF16 is implemented in little-endian encoding: **Microsoft** Code Page 1200.
- On a UNIX-based system, UTF16 is implemented in big-endian encoding: **IBM** Code Page 1200.

You can contact our Support department for more details.

Using Colors in Browser Tasks

To properly use colors on IBM i, you must define all the colors that are used as non-system colors. The easiest way to do this is to access the color file in the uniPaaS Studio and define the colors accordingly.

Browser Client

See the uniPaaS Help file for information about working with Browser Client programs.



Server Initialization

As of uniPaaS V1.8SP1, there is a new command available for starting server engines: **STRUNISVR**. STRUNISVR is the only supported interface for starting uniPaaS server engines on the IBM i.

Please find below a list of server initialization parameters:

Parameter	Values	Type	Description
StartApplication	*INI, Path	Char 256	Project path
Environment File	*DFT, Path	Char 256	Path of the environment file (MGENV400)
INI File	*DFT, Path	Char 256	Path of the ini file
INI Modification (@) File Path	*NONE, Path	Char 256	Path of the @ file
Activate Requests Server	*Yes/*No	Char 4	Connect to broker
Server Instances to Start	Number 1 - 999	Dec 3	Number of engines to start
Log File Path: Name Directory			Path of the log file. No path validation is required.
ANSI CCSID	*DFT, Name	Char 5	Code Page of the IFS files
EBCDIC CCSID	*DFT, Name	Char 5	PASE CCSID
User Parameters	*NONE	Char 500	User string. This string is forwarded to the SBMSVRJOB CL program where it can be parsed and used for defining parameters of the SBMJOB, such as JOBNAME and *JOBQ.
Additional Parameters: 20 sets of three parameters			
INI Section	Name	Char 50	Name of the ini section
INI Entry	Name	Char 50	Name of the ini entry
Value	Name	Char 256	Value of the entry
Load JVM	*Yes/*No	Char 4	Load Java VM during initialization of the server instance



Open Source Exit Programs

uniPaaS for IBM i includes three open source exit programs where you can change the default settings of the uniPaaS server job.

1. **SBMSVRJOB** – An Exit program where the SBMJOB command is executed. By changing/adding exit program logic, you can control all of the SBMJOB parameters, such as Job Name, Job Queue, and Schedule Date/Time.
The sources of this Exit program are located in the **SVRLIB/MGCLSRC** source file, where **SVRLIB** is the name of the server library.
2. **SVRJOBSTR** – An Exit program executed before loading the PASE environment. It is executed from the server job just before the load of the uniPaaS engine. It is useful for the development of additional error handling, settings of additional parameters and environment variables, sending system messages, etc.
This Exit program has no source. It is empty by default.
3. **SVRJOBEND** – An Exit program executed after loading the PASE environment. It is called immediately after the uniPaaS engine is terminated while the IBM i server job is still active.
It is useful for the development of additional error handling, collection of performance information, sending system messages, etc.
This Exit program has no source. It is empty by default.

Server Log File

The uniPaaS log file is created for each server instance. The log file name is set via the **ExternalLogFileName** entry in the MAGIC.ini file. The file path is absolute: **/New Library name/logs/LogFileName**.



New Features

1. New installation method via INSMSEPRD

This method lets you transfer the uniPaaS installation file into the IBM i server via IFS network share, FTP, IBM Client Access or NFS share.

2. New parameter in STRUNISVR: Load JVM *Yes/*NO

The Load JVM parameter lets you disable the load of JVM and therefore eliminates the requirement of having IBM i Java license programs installed on the system.

3. Start uniPaaS engine via MGRB.INI

You can now load uniPaaS server engines during the initialization of the uniPaaS broker. For example, by setting the following entry in the MGRB.INI you can load three uniPaaS server engines with different parameters.

```
[APPLICATIONS_LIST]
Port = STRUNISVR INIFILE('etc/TEST.INI') ACTRQSSVR(*YES) INSTCNT(3),UNIPAAS19,,,1,0
```

Three uniPaaS engines will start from the **UNIPAAS19** installation using **etc/TEST.INI**.

4. Deploying RIA

To be able to deploy a Rich Client application on IBM i platforms:

1. The following files and folders are created once you use the Rich Client Deployment Builder:

```
appname\application
appname\appname.publish.html
appname\uniRC_x_y_z_www\ appname\Images\
```

x,y,z represents the uniPaaS version and www is a unique number representing the specific version

a. Place them in the **uni19RIAApplications/appname** alias on the Web server.

b. Users can access the application from the following URL:

<http://appserver/uni19RIAApplications/appname/appname.publish.html>



2. Add the following into the **httpd.conf** Apache configuration file in this order:

```
AddType application/x-ms-application .application
AddType application/x-ms-application .manifest
AddType application/octet-stream deploy
AddType application/x-msdownload .dll

AddHandler default-handler .jpg .gif .js .txt .bat .msi
```

3. Manually change the **HTTPCompressionLevel** in the application's **publish.html** file to **None**, since there is no compression when the server is a non-Windows platform.

For example:

```
<body onload="initialize()">
<xml id="rcExecProps">
<properties>
<property key="protocol" val="http"/>
<property key="server" val="SYSTEM:PORT"/>
<property key="requester" val="/ServerDir/mgrqcgi019"/>
<property key="appname" val="frame"/>
<property key="prgname" val="START"/>
<property key="arguments" val=""/>
<property key="envvars" val=""/>
<property key="UseWindowsXPThemes" val="Y"/>
<property key="HTTPCompressionLevel" val="None"/>
<property key="DisplayStatisticInformation" val="N"/>
<property key="InternalLogLevel" val=""/>
<property key="InternalLogFile" val=""/>
<property key="InternalLogSync" val="Session"/>
<property key="LogClientSequenceForActivityMonitor" val="N"/>
</properties>
</xml>
<table align="center">
```

Where

SYSTEM:PORT - System Name. Example: "POWER6.MSE.CORP:8080"
ServerDir - Installation directory. Example: UNIPAAS19

5. Performance

There is an overall performance improvement in uniPaaS 1.9i compared to uniPaaS 1.9 and uniPaaS 1.9g (tested on the POWER6 and POWER7 servers).



Licensing

As of uniPaaS V1.8 SP1, there is no need to install **Easycom** licenses. The new license mechanism for the IBM i server is identical to Windows and UNIX. It is based on the **license.dat** file and license broadcasting. The valid license names for this version are:

1. **MGPART2** for the uniPaaS Partition Server
2. **MGENT2** for the uniPaaS Enterprise Server
3. **MGRIA** for the uniPaaS RIA Server

License errors are shown in the server log file.

The following are recommendations regarding licensing:

1. Activate server logging during server configuration and initial tests.
2. Make sure that you have a valid license file set in the **LicenseFile** and the **LicenseName** entries of the **MAGIC400.ini** file.
3. If you have multiple installations of the uniPaaS server on your system, maintain all the licenses in one directory.

Performance – Managing Workload

uniPaaS applications are CPU intensive. Performance tuning of a CPU intensive application on an IBM i platform has always been a complex issue due to the lack of CPU control tools in OS/400 and OS I. CPU intensive applications allocate the maximum number of available CPU resources during runtime, which leads to a loss of performance in other applications. As of OS I 6.1, it is possible to set up a better application isolation at the CPU level. It is applicable for multi-core systems where it is possible to limit the processing requirements of a workload to a subset of processor cores. For more information, see:

<http://publib.boulder.ibm.com/infocenter/iseriess/v7r1m0/index.jsp?topic=%2Frzaks%2Frzaksworkloadcapping.htm>.

Recommendations

Back up all of the uniPaaS configuration files, such as **MAGIC400.INI**, **MGRB.INI**, and **MGREQ.INI**, before modifying them.



Fixed QCRs

727610 – An unmonitored exception occurred during server termination: CPF180B unmonitored by MGCSVR: Command *LIBL/RCLACTGRP not safe for a multithreaded job.

615281 – Temporary files remained after the creation of spool files.

722123 – The mgconst.ger file was missing.

Limitations

1. Connectivity to messaging servers via JMS is not supported using the provided messaging component in this version.
2. The Broker Balancing feature is not supported in this version.
3. Encrypted Communication is not supported in this version.
4. If there is abnormal engine termination (ENDJOB, IPL, ENDSBS), there might be some spool files left by the server engine. The spool files are produced in the OUTQ of the server job user. To make sure that all these spool files are eventually removed from the system, set the OUTQ parameter of the relevant user profiles to the OUTQs that are included in the cleanup routine. For example: QUSRSYS/QEZJOBLOG.

Thank you,

The uniPaaS IBM i Team