Systems Management Development Centre

JADECare Systems Manager

Installation & Administration Guide

Relative to JSM Version	:	4.1.02
Document Date	:	May 2022
Document Status	:	Production Version

Jade Software Corporation Limited cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages, or loss of profits. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Jade Software Corporation Limited.

The information contained herein is subject to change without notice. Revisions may be issued to advice of such changes and/or additions.

Copyright © 2020-2022 Jade Software Corporation Limited.

All rights reserved.

JADE is a trademark of Jade Software Corporation Limited. All trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.

INTRODUCTION 11

JADECARE SYSTEMS MANAGEMENT SOFTWARE 11
JADECare Systems Manager 12
JADECare Systems Agent 12
JADE Application Management Interface 12

INSTALLATION AND CONFIGURATION 13

HARDWARE REQUIREMENTS 13
SOFTWARE REQUIREMENTS 13
TCP PORTS 13
NAMING & DIRECTORY STANDARDS 13
Customer Id 14
Environment Id 14
Application Directories 14
INSTALLING THE JADECARE SYSTEMS MANAGER 16
System Load 16
Starting the System 16
Initial Configuration 16
INSTALLING THE SNMP TRAP DAEMON 17
INSTALLING THE JADECARE SYSTEMS AGENT (JSA) 18
Windows Hosts 18
Linux Hosts 18
INSTALLING THE JADECARE SYSTEMS MANAGER THIN-CLIENT 18

JADE INITIALIZATION FILE 19

OVERVIEW 19
[CNWREFRESH_<ENV NAME>_<SERVER NAME>] SECTION 19
PartitionRetention_<mapfile> 19
PartitionVolume_<mapfile> 19
[JADESERVER] SECTION 20
ServerApplication1 20
[APPENVIRONMENT] SECTION 20
KarmaFileAccessUserId 20
VerifyDatabaseJournals 20
DefeatDownLoadParameterUpdate 21
[JADE] SECTION 21
ShowUserInterrupt 21
[JADECLIENT] SECTION 21
AutomaticCacheCoherency 21
TerminateProcessOnDisconnect 21
[JADESERVER] SECTION 22
AutomaticCacheCoherency 22
[JADERPS] SECTION 22
JSMDeploymentAlterScriptOverride 22
[JADESYSTEMSMANAGER] SECTION 22
AllowCrossCustomerDeploys 22
AllowJariSelectNotCompleted 22
ApplicationIcon 22
AutoDeployProductionOverride 22
AutodeployRequiresAuthorization 23
CentralRepositoryDirectory 23
ComClientUsesAES128Encryption 23
ComClientUsesPort50056 23
CsvExtractSuppressCriteria 23
DisableXCFLogMonitoringDuringDeployment 24
DisasterRecoveryDirectory 24
HandleCountAlertThreshold 24
IgnoreSavedWindows 24
JadeMasterTerminalRootDirectory 24
JSMDeploymentDirectory 24

MasterEnvironmentDirectory 25
OperatorAliasPrefix 25
OperationsRoleId 25
OSPatchHistoryDirectory 25
ProductJADECareFilesDirectory 25
ProductRegistrationDirectory 26
ProductRepositoryDirectory 26
ReportOutputDirectory 26
SNMPTrapDaemonRequired 26
SNMPTrapDaemonService 27
WindowsAuthenticationOnly 27
[JSMDATABASEBACKUP] SECTION 27
BackupDirectoryPermissions 27
BackupUsesMountPoints 28
DisableArchivalRecoveryAlerts 28
ExpectedElapsedTimeFactor 28
OmitPreInstallDbBackupCrcCheck 28
OmitPostInstallDbBackupCrcCheck 28
OnLineBackupResponseInterval 29
ODBCBackupCompressionPercent 29
CommitCoherentBackupMaxTries 29
CommitCoherentBackupRetryInterval 30
[JSMDATABASECOMPACT] SECTION 30
OnlineCompactResponseInterval 30
[JSMDEPLOYMENT] SECTION 30
CreateConfiguration 30
DeployUseJadloadbInterface 31
ExcludeFile<n> 31
Include_doCopyAltosFileFromStagingDir_MiscApps 32
JariEnhancedVersionCheck 32
MigrationVolume 32
RemoveFile<n> 33
SDSCheckTimerDefaultDelay 33
UseReplayableReorg 33
[JSMSTATS] SECTION 34
ClientTranStats 34
LogLogicalTransactions 34
SystemTranStats 34
[KARMA] SECTION 34
ClassesUsingLDAP 34
ClientShutDownDelay 34
CommanderUsingNamedPipes 34
DeployOnlineForceTransitionError 35
DeriveLocalIpAddress 35
JossIPAddress 35
KarmaControl 35
KarmaFileAccessMode 35
MasterCustomerId 36
MasterCustomerName 36
MasterDomainName 36
MasterNTDomains 36
ProcessMonitoring 36
SystemBusySentinelFile 36
SystemType 37
TestMode 37
UpdateReorgWorkDirectory 37
UseLDAPService 37
UseLocalIpAddress 38
[LDAP] SECTION 38
LDAPServer 38
[ODBCMONITORING] SECTION 38

LogLevel 38
[PAGINGGATEWAY] SECTION 38
PagerServer 38
[SMTPSERVER] SECTION 38
EMailDomain 38
[JADEEXECUTEFLAGS] SECTION 38
Joss_Discovery_DisplayFullBuffer 38
Joss_Discovery_IgnoreLastBuffer 39

JSM USER INTERFACE 40

Log On 40
Hot Keys 40
Tool Bar Icons 41
Production vs Test 41

SETTING UP FOR JADECARE ENVIRONMENT MANAGEMENT 42

INTRODUCTION 42
SYSTEM GLOBAL OPTIONS 42
AWS Sheet 42
Backup Sheet 43
Data Retention 44
Product Registration 45
Deployment 47
Environment 47
Host Monitoring 49
Misc 50
Reporting 51
Scheduling 52
Time Zone 54
ROLE SECURITY SETUP 55
JSM USER SETUP 57
CUSTOMER DEFINITION 63
Access Profile References 64
Tracker Statistics 65
LOCATION DEFINITION 66
ENVIRONMENT MAINTENANCE 67
Environment Definition 67
Application Maintenance 69
Deployment 69
Distributions 71
History 72
Thin Client Details 72
Managed Environments 73
AUTO DEPLOYMENT MAINTENANCE 74
SNMP TRACKER INTERFACE 74
Server Host – 900 75
JADE Environment – 901 75
ODBC Environment – 902 75
Parameters 75
Validation 76
Tracker plus Alert 76
TRAPS & MIBS 77
TCP/IP ADDRESS HANDLING 78
Machine Maintenance Summary 80
Net Device Summary 81
SERVER DEFINITION 86
Fully Qualified Domain Name 86
Creating or Updating Host 87
Hardware Details 88
Cluster 88
OS Patch Status 89

Connection History 90
Managed Environments 91
Scripts 92
Tasks 93
Discovery 94
JADE ENVIRONMENT MAINTENANCE 95
Basic sheet 95
Alarms & Exceptions sheet 97
Application Monitoring Sheet 99
Database Structures sheet 103
Miscellaneous sheet 104
Backups Sheet 107
Compact Sheet 108
Reorg Info Sheet 109
Restore Sheet 110
Cluster & SDS sheet 115
Thinclient Sheet 116
Task Sheet 117
SETTING UP FOR SDS ENVIRONMENTS 118
JADE ENVIRONMENT DEFINITION 118
SSL Certificates 118
GLOBAL PARAMETER FILE 119
GlbEnv.txt and GlbODBC.txt details 123
SERVICES DEFINITION 125
ALERT DEFINITIONS 126
Alert Set 127
Definition of an Alert Set 128
Usage of an Alert Set 128
Definition of Performance Items 129
Ignore Logical Disk Volumes from Performance Alert Checks 130
APPLICATION BACKUPS SETUP 130
SYSTEM CONFIGURATION MANAGEMENT SETUP 130
Product Master Repository 130

SETTING UP SINGLE USER APPLICATION SERVER 144

CONFIGURATION REQUIREMENTS ARE AS FOLLOWS: 144
JAWS CONTROL FILE CHANGES 144

SETTING UP FOR ODBC ENVIRONMENT MANAGEMENT 145

INTRODUCTION 145
'USER DATABASE' BACKUP 145
'SYSTEM DATABASE' BACKUP 145
ODBC ENVIRONMENT DIRECTORY STRUCTURE 146
SQL SERVER SECURITY REQUIREMENTS 146

SETTING UP FOR AMAZON WEB SERVICES BACKUPS 147

INTRODUCTION 147
PREREQUISITES 147
AWS CUSTOMER 147
AWS SNAPSHOT 148
AWS SNAPSHOT SCHEDULING 148
AWS SNAPSHOT SCHEDULE OVERVIEW 150
AWS SNAPSHOT SUMMARY 151

SYSTEMS MANAGEMENT ACTIVITIES 152

DELETION OF JSM ENTITIES 152
Jade Environment 152
Master Environment 152
Server Host 152
Cluster Host 152
LICENCE KEY VALIDATION 153

MANAGED BACKUPS 154
Journaled Backups 154
Defining Managed Backup Parameters 155
View Progress 160
Backup Schedule 161
Backup Uses Mount Points 164
ODBC Backup Script Management 165
ODBC Backup Script Requirements 166
Recognized ODBC Backup Script Error Codes 166
Multiple Backup Volumes 167
Compaction 167
Retention of backup directory 167
Compressed Directories 168
Transaction Journals 168
Include External Directories 168
Exclusion of Directories, Files, File Types 169
Backup Sentinel Files 171
Successful Backup 171
Environment Shares 172
Offline Backup Considerations 172
Backup Directory Permissions 172
Backup Directory Shares 173
Backup Scripts 174
Housekeeping 175
Backup History 176
VersionInfo.txt 176
Misc File Copy Errors 176
Erroneous directories ignored 176
Successive Backup Failures 177
MANAGED RESTORES 178
Restrictions 178
Initiation of a Restore 178
Start Restore 180
View Progress 180
Terminate Restore 180
Restore Processes 181
Restore and Directory Structure and Permissions 181
Database Only Restore 182
Files Automatically Excluded 182
Ini File Specified Exclusions 183
Execute Operations scripts During Restore 183
SCRIPT MAINTENANCE 184
Creating a New Script 184
ODBC Query Scripts 185
JADE Transient Method Scripts 186
Task Scheduling Partner Scripts 186
Identify Tasks and Servers Using a Script 187
SCRIPT CATEGORIES 187
TASK SCHEDULING 187
Creating a New Task 189
Updating an Existing Task 190
Selection of Task Targets 191
Definition of Schedule Details 192
Schedule Definitions 193
Standard Output Handling 196
SYSTEM VIEW 197
Auto-Refresh is ON / OFF 197
Show Disabled Entities 197
Enable Expand/Collapse 197
Cluster Host 197
Server Host 198

Host Environments 198
Clients 199
Server Host Fault Explosion 199
ALERT MANAGEMENT 201
Unclaimed Alerts 201
Claimed Alerts 202
Alert Details 202
Entering Notes 203
Alert Handling 203
NODE PROCESS MONITORING 207
Global Settings 207
Host Environment Settings 208
Host Environment Node Monitoring 209
Host Environment and Karma Control 210
APPLICATION DEPLOYMENT 211
Overview 211
XCF Log Monitoring During Deployment Assembly 211
Deployment and Reorganization 212
Optional Deployment File Source 213
Maintenance of Central Deployment Repository 214
Nested Deployments 215
Regime Definitions 215
LoadStyle 217
JADE Command Files 217
Version Control 219
JARI Wizard 219
JARI Batch Build 223
Automatic Deployment 230
JMT Deployment 235
JARI List Form 238
Authorizing and Scheduling Deployments 239
JARI Validation Checks 243
Monitoring Deployments 243
Assembly 243
Installation 244
Operational Deployment Hooks 253
Deployment Scripts 253
User Schema Load Scripts 258
JadeAdmin, Autodist, cnwplugin Automatic Updates 259
Environment Shares 259
Installation Backups 260
Abort Installation 260
Kill Deployment 261
Option to Resume a Recovery after an Exception in File Copy 261
Recovering Using Backup 261
IGNORE, ABORT, RETRY, CONTINUE states 262
Logs and Exception Handling 262
IMPLEMENTATION OF DEPLOYMENT REGIME 3 262
Defining Regime 3 262
Deployments Using Regime 3 262
Thinclient Download Check 263
Restore of Regime 2 Host Environment 263
AVOID CONFIGURATION TYPE FILES BEING OVERWRITTEN 264
JADEADMIN DETAILS 265
Automatic Deployment When Changed 265
JadeAdmin and Single User App Server 265
Ini File 'CreateConfiguration' 265
JadeAdmin Optional Script and Settings 265
SETTING UP FOR AUTO-UPDATING DEPLOYMENTS 267
Overview 267
Setup for Automatic Fat Client Update 268

Setup for SDS Secondary & RPS Node Update 268
SUMMARY OF DEPLOYMENT PROCESS FOR SDS SYSTEMS 269
RECOVERY FROM SDS PRIMARY DEPLOYMENT FAILURES 270
Online Deployments 270
Restarting an Installation 271
Abandoning a Deployment 271
Environment Reset 272
RECOVERY FROM SDS SECONDARY DEPLOYMENT FAILURES 272
Aborting SDS Secondary Installations 272
Recovery Procedure After RPS Autoload Failure. 272
Manual Completion Procedure 273
DEPLOYMENT TO RPS 275
DEPLOYMENT OF UPDATES TO JSA 275
PERFORMANCE AND UTILISATION REPORTING 277
Overview 277
Set Up 277
Graphs displayed on the desk top. 277
Host Performance 278
Application Resource Utilisation 278
Transaction Performance 278
SUMMARY LOGS 279
Selecting Logs 279
Display Logs 279
Printing Logs 280
GLOBAL XCF PARAMETERS 281
Global XCF Item Maintenance 284
Global XCF Group Maintenance 285
Global XCF Summary 285
OS PATCH MAINTENANCE 287
Viewing Hot Fixes Applied: 287
Maintaining Hot Fix and Knowledge Base Articles: 288
Reporting on Hot Fixes: 290
ONLINE COMPACTION 291
Immediate Compactions 292
Scheduling Compactions 292
View Compaction Progress 292
Interrupting A Compaction 293
Cancel Scheduled Compactions 293
Compaction History 294
Perquisites for using Online Compaction 295
Additional Online Compaction Info 295
JSM BATCH INTERFACE 295
Application Monitoring Update 295
Change XCF Parameters 296
Check Backups 296
Check Connected 297
Delete JADE Environment 297
Delete Master Environment 297
Delete ODBC Environment 297
Delete Server Host 298
Extract JARI Details for Environment 298
Extract Product Releases 298
Generate JARI Batch Build Files 298
Get Property Value 299
Global Parameter Distribution 299
Jari Status 299
JSM Version Check 300
JMT Comparison Build 300
Server Patch Status 301
Set Up Environment Directories 301
Set Up Jade Environment Directories 301

Shut Down JSM 301
Set Next Deploy Client Distributions 301
Start JARI Batch Build 301
Switch XCF Group 302
Update JADE Environment 302
Validate and Extract Interested Parties 303
JSA BATCH INTERFACE 303
JadeAdmin and Configuration 303
Cycle JSA Server Application 304
Dump Task Schedule 304
Get Property Value 305
Hot Fix Update Product Repository 305
Initialize ODBC 305
Initiate Discovery 305
JADE Environment Is Active 305
JSA Version 306
JMT Compare 306
Licence Info 306
Run Ad-hoc Backup 307
Run Ad-hoc Compaction 308
Request Update Thinclient Files 309
Validate Ini File 310
Run Environment Restore 310
ACTIONS COMMON TO JOSSBATCH AND KARMABATCHINTERFACE 311
Get Property Value 311
KARMATIMECHECKER 312
ODBC ENVIRONMENT 312
ODBC Environment Directory Structure 312
Monitoring 314
ODBC Monitoring Configuration 317
ODBC Free-Form Query 325
ODBC Free-Form Query Activity 327
ODBC Monitoring Summary 330
.NET SERVICES AND WEB INFO DISCOVERY. 331
Turning the feature On/Off 331
Scheduling the Discovery Process 332
Ad-Hoc Discovery Requests 333
Use of JadeCare.cfg File 333
.NET Services Discovery 336
IIS / Web Details Discovery 338
New Forms and Form Changes 339
Timezone Maintenance 353
WEBHOOK ENDPOINT PROCESSING 354
Token Maintenance 354
Script Builder 356
MS WINDOWS AUTOMATIC UPDATES (PATCHING) 360
System Global Options 360
Server Host Maintenance 361
System View 365
Activity Logging 367
SELECT INTERESTED PARTIES 368
JSM STATISTICS 369
DIAGNOSTICS – LOG ANALYSIS 369
Jommsg Log Analysis 370
ProcNotify Log Analysis 370

APPENDIX 1 – JAWS CONTROL FILE 371

Introduction

This document is a work-in-progress, and does not yet purport to be a comprehensive guide to all aspects of the JADECare Systems Manager and its associated applications.

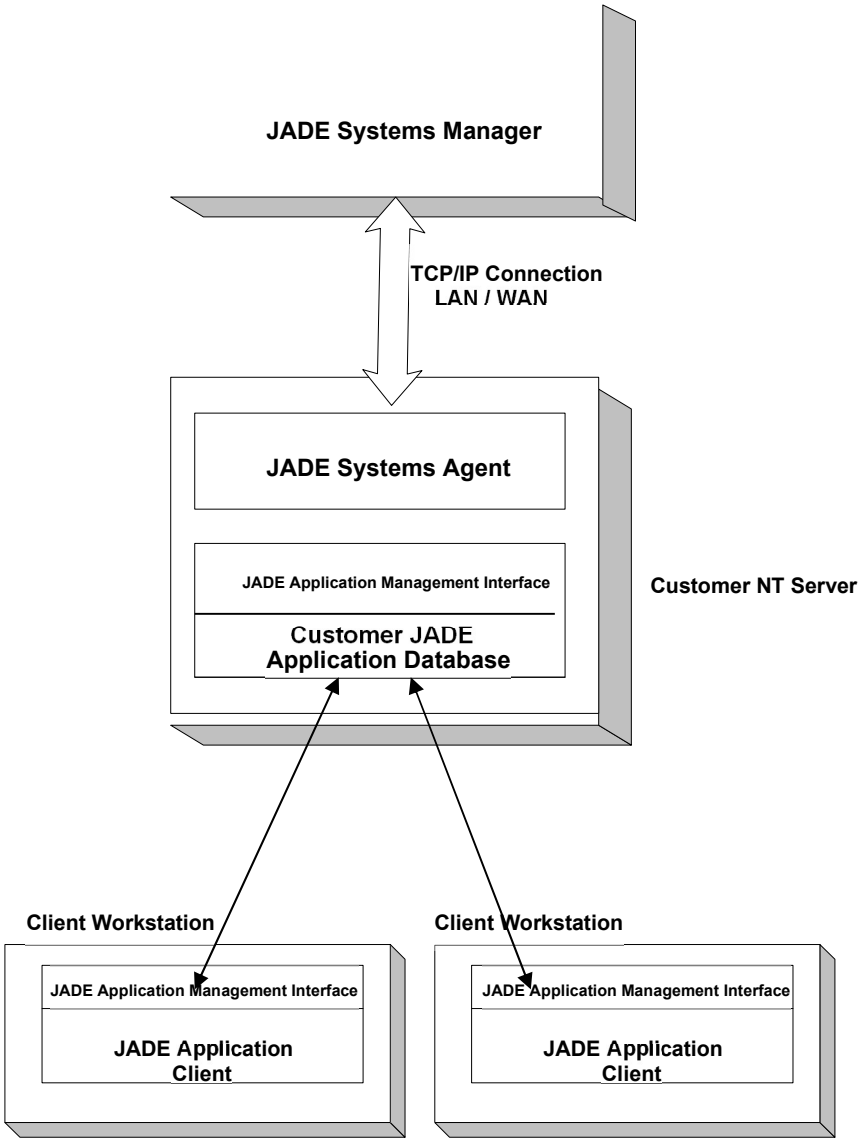
Inquiries about aspects of these products not covered in this document may be directed to jsmsupport@jadeworld.com.

JADECare Systems Management Software

JADECare Systems Management software is a suite of JADE applications management software written by Jade Software Corporation. It consists of three major components, plus a number of purpose-built utility programs. The three major components are the **JADECare Systems Manager**, the **JADECare Systems Agent**, and the **JADE Application Management Interface**.

This document describes the installation and administration of the **JADECare Systems Manager** and **JADECare Systems Agent** only. Other documents describe additional components of the JADECare Systems Management suite.

The following diagram provides a high-level overview of the Systems Management architecture.



JADECare Systems Manager

A single copy of this JADE application runs at your administration centre, and is used to remotely manage all servers that are running JADE applications. Some of the functions it provides are:

- real-time monitoring of key system performance indicators
- automatic alarm generation
- automated deployment and installation
- application backup scheduling and management
- application restore / refresh and management
- application compaction scheduling and management
- monitoring of critical applications
- performance and utilisation reporting

The **JADECare Systems Manager** communicates with the **JADECare Systems Agent** on each of the servers it is managing, via TCP/IP connection using port 12097.

JADECare Systems Agent

A copy of this JADE application resides on each managed server (host). It is responsible for the following activities:

- monitoring the performance of the host machine
- raising alarms to the JADECare Systems Manager when pre-set thresholds are exceeded, or when exception conditions occur.
- reporting of performance and utilisation data to JADECare Systems Manager
- control of backup, restore / refresh, compaction and deployment (installation) processes
- communications with the JADE Application Management Interface of each managed JADE application running on the server (host).
- monitoring of critical JADE applications within each JADE environment

JADE Application Management Interface

All applications that are to be managed by the **JADECare Systems Manager** must inherit this interface, known as **JADECare Start**, and its associated functionality**.** Details of some of this functionality is defined in the **CardSchema Class Library**. This library is provided free of charge by JADE Software Corporation to its customers, and contains support for application management functions plus a variety of classes and methods of general use to application developers.

The main functions of this library that are pertinent to application management are:

- connects to **JADECare Systems Agent** at application startup

- monitors and reports client startup and shutdown
- application version control and deployment
- critical application monitoring
- capture of application performance data
- SDS service monitoring

For more information about this class library, please refer to the **CardSchema Class Library User Guide**.

Installation and Configuration

This chapter describes initial installation and configuration of the **JADECare Systems Manager** (JSM) and **JADECare Systems Agents** (JSA).

Hardware Requirements

Minimum recommended hardware configuration for the JADECare Systems Manager database is:

Dual CPU Server (733 MHZ Pentium III or better)
1 GB Memory
Fast SCSI disks, RAID 0 +1 recommended, with separate disk subsystems for logs and database.
Network Interface Card(s)

Software Requirements

Windows 2008 R2
JADE version and related software - refer JSM release documentation for current information.

If automatic report printing to PDF is required, then the Amyuni PDF Converter is required for the JSM database server.

TCP Ports

In general, sites should avoid use of ports 12090 thru 12099.

The JADECare Systems Agent uses port **12097** to communicate with JSM.

The JADECare Systems Agent also uses ports **12090, 12091, 12093, 12094** and **12095, 12098** for **localhost** communication with other management processes

CNWCommander requires ports **50054, 50055**, and **50056** (see CNWCommander.doc for details).

Managed applications use ports **12095** and **12098** to interface to JSA and may require additional ports, as defined by the developer or your IT Operations staff.

Naming & Directory Standards

The **JADECare Systems Management** environment is designed based on a number of naming standards. These standards must be observed in order for correct functioning of the software.

Customer Id

The **JADECare Systems Manager** is designed to support JADE systems managed for multiple customers. Customers are defined in the **Customer** class in the JSM database. The primary key to collections of customers is the three-character customer id.

Environment Id

Each managed JADE database is referred to as a **JADE Environment**, i.e. an instance of an **Environment**, running on a server. An environment may exist on more than one machine i.e. the same JADE application software may be running in multiple databases on multiple hosts. The environment identifier must be formed according to the following standards:

- 3 characters customer id
- Up to 17 characters free format
- Max 20 characters

Suggested standard **CCCPXXXXX,** where

CCC = Customer Id

P = Production, **U** = UAT, **D** = Development, **T** = Test

XXXXXXXXXXXXXXXXX = Application identifier

Examples : **TESUEXAMPLE**, **TESPEXAMPLE**

NOTE

The 20 character maximum is only available for environments using CardSchema 4.0.02 or later. Earlier releases will only allow a 9 character name.

Environment id **must** include the string "**joss**" for JSM and "**karma**" for JSA.

Application Directories

The JSM / JSA and CardSchema management functions rely heavily on the observance of standard directory naming conventions. These standards are as follows.

Base Directory (<env>)

The base directory (referred to hereafter as **<env>**) takes the form of

<drive letter>:\<environment id>

e.g.

F:\SKIUCASTL

Logs Directory

Normally all logs are placed in **<env>\logs**. For servers that house the JADE database, this should be on a different physical drive from the database itself, e.g.

Z:\SKIUCASTL\LOGS

Normally, most application-related logs will be written to this directory, with database transaction logs written to the CURRENT subdirectory e.g.

Z:\SKIUCASTL\LOGS\CURRENT

Caution

We strongly advise against the use of networked (mapped) drives for database files or transaction logs.

Database Server Directories

The following set of directories is described as the **current** directory set. These directories can be created automatically using the JADECare Systems Manager. The directories are as follows:

<env>\SERVER\C_BIN
<env>\SERVER\C_SYSTEM
<env>\SERVER\C_MISC

<env>\CLIENT\C_BIN
<env>\CLIENT\C_SCHEMA
<env>\CLIENT\C_MISC
<env>\ALTOS

If JADE thin clients are in use, then the following additional directories are required:

<env>\CLIENT\THINCLIENT\C_BIN
<env>\CLIENT\THINCLIENT\C_BIN\DOWNLOAD
<env>\CLIENT\THINCLIENT\C_MISC
<env>\CLIENT\THINCLIENT\C_MISC\DOWNLOAD

All the above directory structures are replicated to provide equivalent **next** and **old** directories, e.g.

<env>\SERVER\N_BIN
<env>\SERVER\O_BIN
<env\SERVER\N_SYSTEM etc etc

Application Server Directories

If the JADE application server is running on the same machine as the database server, then it will use the same directories (binaries and misc) as the database server.

If the application server is running on a different host from the database server then the following **current** directory structure applies:

<env>/C_BIN
<env>/C_MISC
<env>/ALTOS<env>/THINCLIENT/C_BIN
<env>/THINCLIENT/C_BIN/DOWNLOAD
<env>/THINCLIENT/C_MISC
<env>/THINCLIENT/C_MISC/C_MISC

The **next** directory structure is subtly different:

<env>/N_BIN
<env>/N_MISC<env>/THINCLIENT/N_BIN
<env>/THINCLIENT/N_BIN/DOWNLOAD
<env>/THINCLIENT/N_MISC
<env>/THINCLIENT/N_MISC/DOWNLOAD

No **old** directory structure is required.

JADE Initialization File

In the JSM managed environment, the jade.ini file must always be renamed to **<environment id>.ini**, and be located in the **c_bin** directory. Any Windows shortcuts for JADE applications need to include an **ini=** argument in the command line.

Installing the JADECare Systems Manager

Prior to installing **JADECare Systems Manager**, install and register a copy of the required version of JADE on the platform to be used for JADECare Systems Management.

System Load

Load supplied **CardSchema** schema files

Load supplied **JSM** schema files

Add the sections from the **jsm.ini** file (supplied with release) to your standard jade.ini file, rename it according to the above standards, and update entries (e.g.paths, TCP ports etc) to reflect your host environment.

Starting the System

Normally, the JSM database and any associated application server(s) will be started by JAWS via operator command using CNWCommander/ComClient. This procedure assumes that Commander and JAWS have already been installed, and that JSM has been installed as a Windows service.

The primary background application that must be running for JSM is **JossCommsMgr** and is launched at startup of any other JSM application.

All JSM background applications are run as individual nodes as defined in the JAWS control file.

For an example of the JSM JAWS control file see Appendix 1.

JossCommsMgr, when run as a server application, starts **JossSnmpMgr** which is required to receive traps from the SNMP daemon.

Starting the user interface to the JADECare Systems Manager requires a Windows shortcut that runs jade.exe with arguments **schema=Joss app=Joss** (exact setup varies according to use of fat or thin client).

Initial Configuration

This subject covers steps that need to be performed before JSM is ready for general use.

Initializing JSM

Place the **jsm_menu_extract.txt** file from the JSM release into the **<env>\server\c_misc** directory (database server).

Bring up the development environment in Single User mode and execute **JadeScript::initializeJSM**.

This will create the root object singleton, a raft of required objects for several classes and force a load of the JSM and Tracker menu items from the file **jsm_menu_extract.txt** above. This is indicated by a pop-up of how many menu items were read and loaded '<xxx> menu items have been successfully loaded.'.

First User Logon

Once the database is running and JossCommsMgr is running, start a JSM foreground client (app=Joss). When the log on screen is displayed press LogOn and follow prompts. Your user id will be assigned SUPERUSER status. The user id must have the same customer id prefix as the **MasterCustomerId** defined in your ini file. Once you have entered your password, log off and log on again.

Loading JADE Skins

Copy the file **JadeSkin_JSMDefault** from the JSM release to disk on your workstation.

From the JSM menu select **File** then **Import Skin File**, and select the file you have just copied to disk.

Loading the MIB Tree

Copy file **MibData.bin** from the JSM release to your hard disk.

From the JADE Development Environment, select **JadeScript::mibTreeLoad** and follow instructions.

Installing the SNMP Trap Daemon

See Joss Alerts doc.

From release 3.5.02, the trap daemon service can be specified as being required or not and, if required, the service name needs to be specified and will be started and stopped when the JossSnmpMgr application starts and stops.

If the SNMP Trap Daemon is not required, this will avoid the generation of alerts as follows:

```
Emergency Alert for JADECare Systems Manager on <computer name>

SNMP trap daemon 'snmptrapd.exe' is not running/responding on the
JSM server.
No traps can be received while this is out-of-service.
```

This introduces two new ini file options under the **[JadeSystemsManager]** section:

SNMPTrapDaemonRequired=<true | false>
- this item is mandatory.

SNMPTrapDaemonService=<service name>
- required if **SNMPTrapDaemonRequired=true**

At **JossSnmpMgr** start up, if the ini file option **[JadeSystemsManager] SNMPTrapDaemonRequired** has not been specified then an alert will be generated

```
SNMP TrapDaemon Required not specified

Inifile option [JadeSystemsManager] SNMPTrapDaemonRequired value
not explicitly specified, set up error
```

If the service is specified as being required, then the service name must be provided. If it is not defined then an alert is generated:

```
SNMP TrapDaemon Service Name not specified

Inifile option [JadeSystemsManager] SNMPTrapDaemonService has not
been specified but SNMPTrapDaemonRequired=true, set up error
```

If the SNMP Trap Daemon service name is provided it is validated at JossSnmpMgr start up and if not valid an alert will be generated.

```
SNMP TrapDaemon Service Name is not a valid service

Inifile option [JadeSystemsManager] SNMPTrapDaemonService <service
name> was specified but is not a valid service name, set up error
```

Installing the JADECare Systems Agent (JSA)

A copy of JSA must reside on each managed host. JSA is a JADE database with the CardSchema and Joss schemas loaded.

Windows Hosts

For Windows, you can create a master JSA on a suitable host, and then replicate it to all managed hosts. Install a new JADE environment with the standard JADECare directory structures and load CardSchema, JossModel and JossControl schemas. Tailor the ini file as required. This entire system (binaries and database files) can then be copied to each managed host. Minor changes may be required to each ini file, and JAWS setup will be required.

Linux Hosts

The generation and installation of JSA on Linux hosts is semi-automated. For current information, please contact jsmsupport@jadeworld.com.

Installing the JADECare Systems Manager Thin-Client

Some users have both a JSM and Tracker thin-client application running on their desktop and, by default, these show up in the task bar using the same icon. With this release, the thin-client user can define the application icon they wish to use by placing an icon file (suffixed .ico) into the thin-client binaries directory and defining this name in the thin-client ini file **[JSM] ApplicationIcon=<icon file name>**.

Note that the path should not be defined.

JADE Initialization File

Overview

The initialization file for the JADECare Systems Manager (JSM) and JADECare Systems Agent (JSA) must reside in the **c_bin** directory and is to be be **titled <environment>.ini** where <environment> is the environment name.

The following descriptions only cover those initialization file options that are unique to JSM and JSA. JSM and JSA also use many of the standard CardSchema options that apply to all applications that run as subschemas of CardSchema. For information about CardSchema initialization file options, please refer to the Appendix in the **CardSchema User Guide**.

You will see the terms KARMA and JOSS used in various places. These were code names used during JSM development. JOSS was an acronym for JADE Operations Support System, and KARMA an acronym for Kernel And Remote Management Agent.

[cnwrefresh_<env name>_<server name>] section

This section is primarily used for the CNWRefresh tool, a JADECare 24 Operations specific utility, but has impact when using the JSM host environment Restore function.

PartitionRetention_<mapfile>

Default : -1

This keyword can be repeated any number of times with a new mapfile name. and specifies the number of partitions that are to be restored.

Where this is specified, then the most recent number of partitions for the mapfile named will be restored.

The following example would mean that partitioned mapfile **Documents** would only restore the most recent 5 of its partitions.

```
PartitionRetention_Documents=5
```

The following example would mean that partitioned mapfile **Images** would not restore any of its partitions.

```
PartitionRetention_Images=0
```

The following example would mean that partitioned mapfile **Events** would restore all of its partitions that were in the backup.

```
PartitionRetention_Events=-1
```

PartitionVolume_<mapfile>

Default : Uses the database directory specified for this structure in the **backupInfo** file created by the database backup. If this database directory is empty, then the default system path defined for this host environment will be used.

This keyword can be repeated any number of times with a new mapfile name. and allows a volume id to be specified creating a directory location :<volume>:\<env>\server\c_system

Where this is specified, then any partitions restored, the partition control file (<mapfile>.dat) and partition index file (<mapfile_ndx>.dat) will all be located on the volume specified

The following example would mean that partitioned mapfile **Documents,** its index file and any partitions restored will reside on volume T in directory t:\<env>\server\c_system

```
PartitionVolume_Documents=T
```

[JadeServer] section

ServerApplication1

For JSM database nodes, this option should be specified as

```
ServerApplication1=Joss,JossCommsMgr
```

For JSA database nodes, this option should be specified as

```
ServerApplication1=Joss,KarmaCommsMgr
```

For JADECare managed database nodes, this option should be specified as

```
ServerApplication1=CardSchema,KCCMgr
```

Note that for SDS secondary and RPS nodes using deployment regime 2, this statement is **mandatory**, i.e. KCCMgr must be the first server application launched.

[AppEnvironment] Section

KarmaFileAccessUserId

Note that this option is placed in the ini file belonging to the *managed environment*, not the JSA's ini file. It defines the userid and password to be used for connection to JSM's FTP server when downloading deployment files.

KarmaFileAccessUserId=<userid>/<password>

Default value = **anonymous**

Note that any password supplied is suffixed by JSA with a secret character string to prevent use of the password by someone who obtains unauthorised access to the .ini file definition. This character string will be advised upon request from need-to-know users (i.e. the administrator who sets up the FTP server accounts.)

VerifyDatabaseJournals

Boolean value, true or false, default value = true

When this value is true, then each time a database journal is switched an application called KCVerifyJournal is started to verify the integrity of the journal.

If the verification of the journal fails, an alert will be generated to JSM with short description **EMERGENCY ALERT : Database journal verification failed on <host> for <env>** and a long description describing the error and that the system should be shut down immediately and an offline backup taken.

NOTE: That if this value is true, then the **PersistentDb** section **JournalCloseAction** must be one of

- **none**
- **copy**
- **copyandcompress**

Any other value will cause an alert to be generated with a short description of **Disallowed JournalCloseAction ini file setting for environment <env> on <host>**.

DefeatDownLoadParameterUpdate

During deployments that include thin client files, JSA normally updates the download parameters in the [**JadeAppServer**] section of the ini file (**DownLoadVersion**, **DownLoadDirectory**, **DownLoadDirectory2**). These updates can be defeated by including the entry

[AppEnvironment]
DefeatDownloadParameterUpdate=true

This option defaults to a value of **false**.

[Jade] Section

ShowUserInterrupt

Needs to be defined as false for all presentation clients (thin- and fat- clients) to prevent the JADE User Interrupt from displaying on the desk top. The User Interrupt allows:

- Break Application - Interrupt method execution
- Start / stop /reset of profiler
- Trace methods
- Terminate Application

None of the above should be visible for a production application.

[JadeClient] Section

AutomaticCacheCoherency

String value :True, false or serverdefault

Default False.

This value must be set to **true** for 3.0 and later releases of JSM, JSA and Tracker.

TerminateProcessOnDisconnect

String value, true or false

This value should be set to **true** for JSM RAP and JADE.EXE programs that are spawned by the RAP (e.g. JossReportRunner). Setting this option to true ensures that these fat clients close down when the RAP is shut down by JAWS.

[JadeServer] Section

AutomaticCacheCoherency

String value :True or false

Default False.

This value must be set to **true** as the 3.0 and later releases of JSM, JSA and Tracker.

[JadeRps] Section

JSMDeploymentAlterScriptOverride

This ini file controls whether the deployment engine is allowed to update the alter script related ini file settings to true unconditionally.

The alter script related items are:

```
[JadeRps]
AutoRunReorgDropScript
AutoBulkLoadHistoricalTables
DropHistoricalTableOnAddExisting
```

[JadeSystemsManager] Section

AllowCrossCustomerDeploys

Default = false, optional

This option is only to be used where deployments are allowed to use JARIs from environments from more than one customer. It is **NOT** to be used by JADECare 24.

Allows JARIs from all customer environments to be visible for selection when building a JARI.

See also **AllowJariSelectNotCompleted.**

AllowJariSelectNotCompleted

Default = false, optional

This option, when true, allows the selection of JARIs from other host environments that are syntactically complete and are not logically deleted and have yet to be authorised, distributed or actually deployed.

When this option is false, then only JARIs that have been successfully deployed are available for selection by other environments. See also option **AllowCrossCustomerDeploys.**

ApplicationIcon

Default = null, optional

This is applicable only to thin-client initialization files and allows the user to define a specific icon for each JSM and/or Tracker thin-client session and is primarily intended to be able to differentiate between the two applications on the task bar.

AutoDeployProductionOverride

Defunct – as of 3.8.02

AutodeployRequiresAuthorization

Default = false, optional

Be default, an autodeployment is automatically authorized and the deployment will proceed to completion.

Setting AutodeployRequiresAuthorization will halt the auto deployment at the authorization stage of the deployment. Manual authorization is then required to complete the deployment. Note that the deployment then needs to be approved or cancelled before a new deployment can be attempted.

CentralRepositoryDirectory

Default = <null>, required

Read Location : Database Server

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the location where development groups can lodge deployment files for the JSM deployment engine to use when building the deployment package.

Refer to section under **Deployment** titled **Optional Deployment File Source** for further details.

The current value of this directory can be viewed via the **System Global Options** form on the **Product Registration** sheet under caption **Central Repository**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

ComClientUsesAES128Encryption

Default : true, optional

Read Location : thin-client

The latest releases of **CnwComClient** default to using AES 128-bit encryption.

However, in some cases, file copies have used significant amounts of CPU and have incurred significantly longer file copy times. With this option set to **false** **CnwComClient** is executed with **–c** argument set reverting to the RC5 encryption used before the introduction of AES 128-nit.

ComClientUsesPort50056

Default : false, optional

Read Location : Database Server

This option allows the **CnwComClient** process to be started using Port 50056 in those cases where the thin-client is started outside of the same domain and without AES Encryption specified. This starts the **CnwComClient** process with the **–d** argument.

CsvExtractSuppressCriteria

Default : false

Read Location : Thin-client workstation

This option, if true, will cause the selection criteria that prepends the csv extract file to be suppressed.

DisableXCFLogMonitoringDuringDeployment

Default = false

Type : Boolean

This option can be used to allow JSM to advise Commander to stop or start monitoring JSM's internal deployment staging area. When one or more deployments are in the assembly phase, conflicts with Commander can occur should the directory be one of the XCF parameters. This ini file option, if true, allows the first deployment to cause monitoring of said directory to stop and for the last deployment terminating to re-start the monitoring.

DisasterRecoveryDirectory

Default = <null>, optional.

This is a required item for JADECare Software Corporation only.

This directory defines the location where SLA details deemed critical in a disaster is contained. These details are periodically extracted from JSM if the Tracker module has been implemented and the Weekly Extract scheduled item is enabled in the **System Global Options** form on the **Scheduling** sheet

The current value of this directory can be viewed via the **System Global Options** form on the **Misc** sheet under caption **Disaster Recovery Directory**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

HandleCountAlertThreshold

Default 500,000

Applies to JSA ini file. This value defaults to 500,000, however should be set to about 58,000 for Windows NT 4.0 operating systems as the maximum handles allowed for that OS is about 62000.

IgnoreSavedWindows

Default false

If this option is true, then any GUI session that starts will ignore the saved windows information from the previous session. This means that only the MDI form will be presented and any windows that were open when the last session was closed will NOT be opened.

JadeMasterTerminalRootDirectory

Default = null.

This defines the root directory to be used when doing a 'Build and Register' of a JMT product release (base or consolidated) and provides a starting point from where the user can select the directory where the JADE Master Terminal release is to be uplifted from.

JSMDeploymentDirectory

Default = <null>, Required

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the staging area where the JSM deployment engine builds the deployment package.

The current value of this directory can be viewed via the **System Global Options** form on the **Product Registration** sheet under caption **JSM deployment repository**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

MasterEnvironmentDirectory

Default = <null>, Required

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the location where the latest version, thinclient files and other data related to each **Master Environment** is located. This area is used regularly when managed environments are launched, when thinclient checks are done and other internal management functions occur.

The current value of this directory can be viewed via the **System Global Options** form on the **Environment** sheet under caption **Master Environment Directory**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

OperatorAliasPrefix

Default : 'none', optional

This value is prepended to the JSM log on used after having been validated against the NT domain and must match an operator definition within JSM.

This option may have different values for each application server. It must be 3 characters in length and these 3 characters must equate to a valid customer declared in JSM.

OperationsRoleId

Default : CNWCS1, required

This option is used to define the role (user id) to which automatically raised JADECare Tracker faults are to be sent, and to identify the assignee (user id) when the Response Threshold Monitoring generates a Tracker fault (if Tracker has been implemented).

For correct system functioning, this option **must** be correctly set up for all JSM ini files (app servers and RAP) when the JADECare Tracker module is in use.

OSPatchHistoryDirectory

Defines the repository where JSM can find the OS Patch Maintenance files, used by application **JossOSPatchMgr**

ProductJADECareFilesDirectory

Default = <null>, required.

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the location where JADECare supplied files are located which are used as part of the product registration process.

The current value of this directory can be viewed via the **System Global Options** form on the **Deployment** sheet under caption **JADECare Files Directory**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

ProductRegistrationDirectory

Default = <null>, required

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the location where JADE supplied files are located which are used as part of the product registration process.

The current value of this directory can be viewed via the **System Global Options** form on the **Deployment** sheet under caption **CD media and zipped files.**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

ProductRepositoryDirectory

Default = <null>, required

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the location where registered products and their hot-fixes are located which are used as part of the deployment process when upgrading JADE, CardSchema or JADE Master Terminal.

The current value of this directory can be viewed via the **System Global Options** form on the **Deployment** sheet under caption **Product repository.**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled.

Note : Hot Fixes require a sub-directory for each product as all products can now register hot fixes.

ReportOutputDirectory

Default = <null>, required

Required item, generates alert if not specified when **JossGeneralEventMgr** starts.

This directory defines the location where background scheduled report output is to be located.

The current value of this directory can be viewed via the **System Global Options** form on the **Reporting** sheet under caption **Output directory for PDF and CSV files.**

To modify this value within a running JSM, the application **JossGeneralEventMgr** will need to be cycled

SNMPTrapDaemonRequired

Default = <null>, Required

Required item, generated alert if not specified when **JossSnmpMgr** starts.

If false, this suppresses:

- The validation and start up of the SNMP Trap Daemon
- The repeated alerts if the SNMP Trap Daemon service is not active.

SNMPTrapDaemonService

Default = <null>, optional

Required if keyword **SNMPTrapDaemonRequired=true.**

If deemed to be required and item has not been specified, then an alert is generated.

Service name provided is validated when **JossSnmpMgr** application starts up and will generate an alert if the service is not valid.

The service name specified is started and stopped as application **JossSnmpMgr** starts and stops.

SuppressInterimDeploymentEmail

Default : false

When this option is specified as **true**, then the interim email usually issued by a deployment is suppressed.

WindowsAuthenticationOnly

Default : false, optional

When this option is specified as **true**, then the user id and password entered in the logon form is authenticated against the Windows domain and must match for JSM or Tracker logon to be achieved.

When this option is false, then the derived user id (see OperatorAliasPrefix earlier in this section) and password is validated against the Windows domain. If this fails, then the userid and password are validated against the values stored within JSM's database.

[JSMDatabaseBackup] Section

The options in this section are *used* by JSA, but must be in the *managed application's* ini file.

BackupDirectoryPermissions

This defines how and if the backup directory for the host environment will be implemented. Refer to the discussion in the section **Managed Backups** for further information on implementation of permissions on the backup directory.

Default = **Replace**

Valid entries are:

None	No changes to permissions
Edit	Add permissions below to existing permissions
Replace	Replace any permissions defined with those below.

If the entry does not exist or is not one of the above, the default action of Replace is used.

If permissions are added or replaced then the following permissions will be set:

SYSTEM	Full Control
Administrators	Full Control

<environment>	Full Control
<environment>_G	Read access
<environment>_M	Read access

Warning: If None is used, then it is imperative that the permissions defined on the final backup directory for the host environment contain the following as a minimum. Any other setting will cause the backup to fail.

SYSTEM	Full Control
Administrators	Full Control
<environment>	Full Control

BackupUsesMountPoints

This option allows the specification of the backup directory structure to be defined via a script (which will allow one or more directories to use mount points to another volume). This option being set to true means:

- that the backup and deployment will NOT perform any free space checks before initiating the backups
- expects either a script titled **buildBackupDirectoryStructure.cmd** or **buildBackupDirectoryStructure.ps1** to be resident in the <env root>\server\c_misc\cardinal directory.

Refer to section "Managed Backups" sub-section "Backup Uses Mount Points" for more detail.

DisableArchivalRecoveryAlerts

Valid values for this are **true** and **false**. Normally, JSA or JSM will raise alerts if ini file option EnableArchivalRecovery in [PersistentDb] is found to be false. The DisableArchivalRecoveryAlerts option allows you to defeat this alert mechanism.

Default = **false**.

ExpectedElapsedTimeFactor

Each backup's expected elapsed time is calculated from the average of the last five successful backups (kept by online and offline) and incremented by the ExpectedElapsedTimeFactor.

An integer value is expected. Should the item not be present or not be a valid integer then the internal default of 50% is assumed.

OmitPreInstallDbBackupCrcCheck

Valid values for this are **true** and **false**. If this option is set to **true** prior to a pre-installation backup (during deployment) then JSA will not perform a **VerifyCheckSums** operation on the backed up database. Normally it is highly desirable to perform this safety check, however in can be defeated using this option. JSA always returns the value of the option to the default once it has been referenced.

Default = **false**

OmitPostInstallDbBackupCrcCheck

Valid values for this are **true** and **false**. If this option is set to **true** prior to a post-installation backup (during deployment) then JSA will not perform a

VerifyCheckSums operation on the backed up database. Normally it is highly desirable to perform this safety check, however in can be defeated using this option. JSA always returns the value of the option to the default once it has been referenced.

Default = **false**

OnLineBackupResponseInterval

The backup manager process expects responses from the target system's online backup application (KCOnLineBackup) within a specified elapsed time. Failure to obtain a response within this timeframe causes the backup process to consider that there is a problem with the target system and issue an alert. If the condition persists, then repeat alerts will be generated. Should a response subsequently be received, then any alert generated will be marked as cleared.

The time between these responses is dependent on:

- CPU and I/O media speeds
- the size of the map file being backed up
- in the case of online backups, when the backup can obtain the database resources.

The current internal default is set to 180 seconds and for some systems has been shown to be insufficient time.

This new option allows this interval to be set specific to the system being backed up.

The value must be an integer value and must by more than 9 seconds. If the value is not present, not an integer or less than 10 seconds, the default value of 180 seconds is used.

ODBCBackupCompressionPercent

This optional parameter is used by JSA to obtain the compression factor to be used when calculating the expected backup space required for an ODBC backup.

This parameter is applied to ALL ODBC databases being backed up by JSA regardless of the ODBC Instance.

The value must be greater than 10 and less than 100 for the value to be used.

CommitCoherentBackupMaxTries

Type : Integer (1+)

Default : 3 (times)

This optional parameter applies to a scheduled backup of an SDS Native database intended for restore to an RPS node.
A requirement for this purpose is that the backup captures a quiet point on the Primary database after backing up the database files. Because the request for a quiet point can timeout, this setting provides the option to retry the backup commit (and quiet point request).
If all attempts fail then the backup will complete with a standard commit, meaning that the backup is suitable for restore to itself, but not an RPS node.

See also commitCoherentBackupRetryInterval (next).

Note: When taking an ad-hoc backup of an SDS Secondary native database, it is possible to specify that the backup *must* be suitable for RPS restore. In such a

case, the backup will attempt completion indefinitely (or until the backup is aborted).

CommitCoherentBackupRetryInterval

Type : Integer (1+)

Default : 5 (minutes)

This setting works in unison with commitCoherentBackupMaxTries (see above).

This optional parameter applies to a scheduled backup of an SDS Native database intended for restore to an RPS node. When a quite point cannot be achieved on backup commit, this setting specifies the time in minutes between backup commit retry attempts.

[JSMDatabaseCompact] Section

The options in this section are *used* by JSA, but must be in the *managed application's* ini file.

OnlineCompactResponseInterval

This parameter specifies some number of *minutes* within which JSA expects feedback from a managed compact operation (application KCOnLineCompact).

Failure to obtain a response within this timeframe causes the compact process to consider that there is a problem with the target system and issue an alert. If the condition persists, then repeat alerts will be generated. Should a response subsequently be received, then any alert generated will be marked as cleared.

The current internal default is set to three minutes and for some systems this has been shown to be insufficient time.

To avoid unnecessary alerts, set this parameter to the time (in minutes) required to compact the largest database file.

[JSMDeployment] Section

This area is related to application environment deployments.

CreateConfiguration

Type : Boolean

Default value : false

Added 4.0.03 (hot fix 4.0.02.014)

If the database environment is not SDS capable and there are no remote fat-clients associated with the database, then setting this option will cause deployments to execute JadeAdmin with argument 'createConfiguration' rather than 'updateConfiguration'

The 'updateConfiguration' requires access to Domain Controller and Active Directory to gather group membership and other details which is done per file and, on some sites, leads to lengthy processing delays.

NOTE: Do NOT use this option if there are or may be remote fat-clients for this environment.

```
[JSMDeployment]
CreateConfigurtion=true
```

DeployUseJadloadbInterface

Boolean value : true or false.

Default value false.

This option can be specific in the JSM ini file and will force the schema load interface to use jadloadb rather than using KCCMgr and the RootSchema,JadeSchemaLoader application.

The only exception to this will be if there is one or more Jade Environments detected that are defined as Single User Application Servers.

ExcludeFile<n>

String value of <directory>,<true|false>,<file mask>

No default value.

This definition is expected in the ini file for the environment being deployed to and is NOT used if within the JSM ini file.

The exclude file options can be validated by executing JSA's KarmaBatchInterface action=validateIniFile.

The <directory> may specify a specific directory node or the next level below the node. A specific directory node example is d:\example\server\c_misc\apps. To specify the nodes below as the starting point an example is d:\example\server\c_misc\apps*.

The <true|false> part of the string defines whether to parse all sub-directories. True indicates all sub-directories are to be parsed

The <file mask> can define any valid mask that can be applied to a JADE FileFolder instance. Examples are:

- *.json
- appsettings.json

Example:

An example is where a Jade Environment has .Net applications (running under one or more sub-directories in \server\c_misc\apps) and one or more requires a new configuration file named 'appsettings.json'. The ini file settings below would ensure that the appsettings.json in the active directories are not overwritten accidentally by a deployment. See also RemoveFile below.

```
[JSMDeployment]
ExcludeFile1=d:\example\server\c_misc\apps\*,false,appsettings.json
```

The "ExcludeFile1' definition would ensure that any files named 'appsettings.json' found in the first level of sub-directories in d:\example\server\c_misc\apps are not propagated by JadeAdmin to any SDS and/or remote environments.

The example will insert an entry in the [plugin.excludeFiles] for each file in any sub-directory of d:\example\server\c_misc\apps. So if the following structure existed:

```
d:\example\server\c_misc\apps
d:\example\server\c_misc\apps\Example
d:\example\server\c_misc\apps\Example\an_example.txt
d:\example\server\c_misc\apps\Example\appsettings.json
d:\example\server\c_misc\apps\Example\z_anotherfile.html
d:\example\server\c_misc\apps\Express
```

```
d:\example\server\c_misc\apps\Express\an_example.txt
d:\example\server\c_misc\apps\Express\appsettings.json
d:\example\server\c_misc\apps\Express\z_anotherfile.html
```

Then the following entries would be defined in the JadeAdmin.cfg file:

```
[plugin.excludeFiles]
d:\example\server\c_misc\apps\Example\appsettings.json
d:\example\server\c_misc\apps\Express\appsettings.json
```

Include_doCopyAltosFileFromStagingDir_MiscApps

String value of <true|false>

Default value false.

Added 4.1.02 (hotfix 4.1.01.006).

This ini file setting is only valid for environment's flagged as regime 3.

If the ini file item is not explicitly defined in the environment's ini file, the next deployment will ensure that it is explicitly defined with a value of false.

When the value obtained is true, then this will cause any 'createConfiguration' or 'updateConfiguration' (via KarmaBatchInterface) or deployment to ensure that the following is defined in the JadeAdmin configuration file:

```
[plugin.doCopyAltosFileFromStagingDir]
key1=misc\Apps
```

JariEnhancedVersionCheck

String value of <true|false>

Default value : false

Added 4.1.02 (hot fix 4.1.01.009)

This is only valid for those JARIs which need to be manually authorized.

When this option specifies **false** or is not defined then the existing paradigm continues which will cause the JARI authorisation to fail should the JADE and CardSchema versions for registered environments not match those when the JARI was built (ie as the result of a restore).

When this option is **true** then enhanced version checks are performed as follows:

1. verify that the JARI is not deploying more than one major JADE release ahead
2. Check that if JADE releases differ and there is no JADE upgrade defined that there are no JADE hotfixes in this JARI as these would be incompatible with the JADE version being run.
3. Check that if JADE releases differ and there is no JADE upgrade defined that there are no CardSchema changes in this JARI as any binaries would be incompatible with the JADE version being run.

MigrationVolume

Character value.

Default value : the current database volume

This option may be used where a database migration, for example from JADE 6.3.12 to JADE 7.0.10, has insufficient free space for the existing database and the new database created as a result of the migration.

When this option is used, the value differs from the database volument and JSA 3.7.05 has been implemented, the database files are copied to <migration volume>:\<env>\server\c_system<nn> where nn is 63 or 70 depending on the environment's current JADE release. The migration process will use this migration database path as input.

NOTE: The use of this option will have a significant impact on the deployment outage as the copy is single-threaded.

RemoveFile<n>

String value of <directory>,<true|false>,<file mask>

No default value

The remove file options can be validated by executing JSA's KarmaBatchInterface action=validateIniFile.

The <directory> may specify a specific directory node or the next level below the node. A specific directory node example is d:\example\server\c_misc\apps. To specify the nodes below as the starting point an example is d:\example\server\c_misc\apps*.

The <true|false> part of the string defines whether to parse all sub-directories. True indicates all sub-directories are to be parsed

The <file mask> can define any valid mask that can be applied to a JADE FileFolder instance. Examples are:

- *.json
- appsettings.json

Example:

An example is where a Jade Environment has .Net applications (running under one or more sub-directories in \server\c_misc\apps) and one or more requires a new configuration file named 'appsettings.json'. The ini file settings below would ensure that the appsettings.json in the active directories are not overwritten accidentally by a deployment.

```
[JSMDeployment]
RemoveFile1=d:\example\altos\ansi\server\x64-msoft-win64\misc\apps\*,false,appsettings.json
```

The "RemoveFile1' definition would ensure that any files named 'appsettings.json' found in the first level of sub-directories in d:\example\altos\ansi\server\x64-msoft-win64\misc\apps are removed and not propagated by deployment into the active c_misc director or by JadeAdmin to any SDS and/or remote environments.

SDSCheckTimerDefaultDelay

Integer Value (in seconds).

Default value is 5 seconds.

This value is used by the JSA deployment module on an SDS Secondary and defines the wait period between SDS tracking halt checks. When the deployment is at transition then the default value is doubled (ie if default is 5 secsonds then the wait period at transition is 10 seconds).

UseReplayableReorg

String value of true or false

Default value = false

If the system is an SDS primary, then this value will be set to true (if not already) as this is a mandatory requirement of SDS Primary.

This value is used to:

- execute jadloadb with replayablereorg option set appropriately
- during a deploy when a reorg is detected, this option being true will allow an online post-deploy backup to be taken, otherwise the post-deploy backup will be offline.

[JSMStats] Section

This area is also related to application environments set up for statistics gathering.

ClientTranStats

Default value true

Identifies whether or not client transaction statistics is to be gathered for this application.

This item is obtained from the application server's initialisation file. If the value is explicitly set to false, then the thin-client initialisation file is checked.

LogLogicalTransactions

Default value false

When true, each logical transaction encountered has pertinent information written to a log (title format <application name>_ltl_<ccyymmdd>.log)

SystemTranStats

Default value 0

When set to 2, this will cause client transaction statistics to be gathered and forwarded to JSM.

[Karma] Section

ClassesUsingLDAP

No longer used as from 3.1 release

ClientShutDownDelay

Integer value

Default value = 45

Time value in seconds waiting for clients to shut down before starting to 'kill' outstanding processes and advising KCCMgr to terminate itself.

CommanderUsingNamedPipes

Possible values **true** or **false**.

Default=**true**

Defines whether JSA connections to CNWCommander will use named pipes or TCP/IP connections.

DeployOnlineForceTransitionError

Possible values – name of specific JADE Environment (case-insensitive)

Default : null

If a value is defined then every online deployment will check whether the name matches the JADE Environment being deployed to. If a match, then error state is forced with error message '<env name> force error in online deploy transition ([Karma] DeployOnlineForceTransitionError)'.

DeriveLocalIpAddress

Possible values : **true** or **false**.

Default : **false**

Used by JSA only and is only valid with Windows 2008 or later O/S. Allows the local IP address (IPV4 only) to be derived for the best route for the destination address Also refer to setting **UseLocalIpAddress.**

If the O/S is Windows 2008 or later and this keyword has not been defined before, it is created with a value of **true.**

JossIPAddress

String value.

Default : null value

This entry is required in JSA's ini file. Note that the IP address is required rather than the host name, because the host name may not necessarily be correctly resolved if the host running JSA does not have access to the same domain name server (DNS) as JSM.

KarmaControl

Possible values : **True** or **False**

Default = **False**

Required to be True for each JADE managed system and also for JSA and JSM as they are also JADE managed systems.

Note that the above value is derived from the ini file on the application server host if using a JADE thin-client.

KarmaFileAccessMode

Possible values : **FTP, HTTP, HTTPS**

Default = **FTP**

Required by:JSA only.

Identifies the type of file copy mechanism that is to be used for file transfers from JSM during application deployments.

MasterCustomerId

This is a three-character identifier for the master customer i.e. the system manager.

A value must be specified.

MasterCustomerName

This is a descriptive value used to identify the master customer in a number of displayed messages, reports etc. Maximum 30 characters.

Default=**JADE Direct**

MasterDomainName

This value defines the internet domain to which hosts of the master customer (i.e. system manager) belong.

Default = **cnw.co.nz**

NOTE : The first level of the MasterDomainName must be the same as the MasterCustomerId.

MasterNTDomains

This value defines the Windows NT domains that belong to the master customer. If more than one domain is listed, they must be separated by commas, e.g.

MasterNTDomains=DOMAIN1, DOMAIN2, DOMAIN3

Default = **CNW**

The Master NT Domain entities are used to identify whether the server concerned is in a JADECare managed domain or a customer domain when defining the content and distribution of the global parameters.

If the server's domain is in the defined JADECare managed domains then all JSM details are provided in the global parameter files. If the server is NOT in the defined JADECare managed domains then the data contained is restricted to the customer relevant data for the customer that the server is associated with (the first 3 characters of the server name).

ProcessMonitoring

Possible values :**True** or **False**

Default = **True**

This identifies whether the host concerned is to have its environment resource utilisation monitored via ProcNotify.exe and ProcMonitor.sys. Normally ProcNotify runs as a service, and JSA or JSM will raise repeated alerts if this is not present. To defeat such alerting when ProcNotify is not in use, set the value of ProcessMonitoring to false.

SystemBusySentinelFile

String value.

Default value = <environment>\server\c_misc\cardinal\system_busy.ctl

The presence of this file will prevent backups and deployments from stopping this system and causes the backup to terminate immediately and return a status of failed, or a deployment to return to hold state.

SystemType

Possible values : **JSM**, **JSA** otherwise ignored

Default : **<null>**

This value is always derived from the database server application (the jadrap.exe).

Explicitly identifies whether the application is JADECare Systems Manager or JADECare Systems Agent. If the item is not found, the value is null or other than **JSM** or **JSA**, then the database path is interrogated to identify if the system is JSM or JSA (ie. if the path contains either the names 'joss' or 'karma').

TestMode

Possible values : 0 to Max_Integer

Default : 0

This keyword is intended for JSM/JSA development and qualification use and for test versions of JSM/JSA.

For production systems, the TestMode value should not be declared and will then default to zero.

For UAT systems, the test mode level suggested is 1.

Any value greater than zero will cause the system to be a test system, invokes several diagnostic functions and inserts the characters [TEST] into the MDI frame caption. Some of the diagnostics may be dependent on the test mode level declared.

Any value other than zero will enable the use of various sentinel files, for further information on these contact the JSM development group.

If the above option is greater than zero and an icon file titled **JSM_Test.ico** is resident in the database server's server/c_misc directory, then this icon will be displayed on the task bar rather than the default JSM icon.

UpdateReorgWorkDirectory

Possible values : **true** or **false**.

Default : true

When this keyword is set to true, the following environment ini file settings are written at the start of a compact or deploy operation:

[JadeReorg]
ReorgWorkDirectory=<filepath>
ReorgBackupDirectory=<filepath>

This results in work and backup files being created at the database file location (as opposed to invoking copy operations to and from the system directory).

The UpdateReorgWorkDirectory setting is provided so that you may specify your own paths for work and backup directories that will not be overwritten (by setting UpdateReorgWorkDirectory=false).

UseLDAPService

Possible values : **true** or **false**.

Default : **false**

Identifies if applications are to use the LDAP service to refresh common information about staff and roles.

If **true**, then the **LDAP** section should also be defined, see below.

UseLocalIpAddress

Possible values : **<ip address>** or **null**.

Default : **null**

Used by JSA only and with Windows 2008 or later O/S. Allows a specific local IP address (IPV4 only) to be specified when the first interface on the adaptor is not the desired one. Also refer to setting **DeriveLocalIpAddress**

[LDAP] Section

LDAPServer

String value identifying the DNS name of the LDAP service. If the **Karma** section has **UseLDAPService** set to true and the applications attempt to use the LDAP Service, this entry is mandatory. If not provided, then any application that attempts to use the LDAP Service will cause an exception.

Default value : **null.**

[ODBCMonitoring] Section

LogLevel

Integer value. Default – do not define.

Used for additional diagnostics in the cnwsql.dll and logs into the cnwsql.<timestamp>.log

For diagnostics and debugging set this to 4.

[PagingGateway] Section

PagerServer

Identifies the server hosting the Paging Gateway system. This is required for any paging activity implemented by JSM and/or Tracker.

The default value is @pager.cnw.co.nz.

[SMTPServer] Section

EMailDomain

Identifies the email domain.

If using LDAP defined roles, then this item is required as the domain is appended to the role UID to facilitate paging.

The default value is @jadeworld.com

[JadeExecuteFlags] Section

Joss_Discovery_DisplayFullBuffer

Joss_Discovery_DisplayFullBuffer=<true | false>

The **Joss_Discovery_DisplayFullBuffer** option can be used in JSM and/or JSA.

In JSA it will force the logging of the entire discovery buffer returned from the karma.dll entrypoints. Note that this will NOT include the buffer extensions added by JSA for .Net Services from the config file or the web application jadehttp.ini extensions .

In JSM it will log the entire buffer content received from JSA (including the JSA extensions).

Joss_Discovery_IgnoreLastBuffer

Joss_Discovery_IgnoreLastBuffer=<true | false>

The **Joss_Discovery_IgnoreLastBuffer** option is only applicable to JSA and only when [Karma] TestMode=<greater than zero) has been specified.

When false then if the discovery buffer is identical to the last request of this type then no data is passed back to JSM.

When true then the value of the last discovery buffer is ignored and the buffer content is unconditionally sent to JSM

JSM User Interface

Log On

The following form is the log on form required to access JSM.

The above log on can be varied with the use of various initialization file settings (see below).

With the default initialization file settings, the above user id and password is authenticated against the Windows domain. If that fails then a match is tried to Operator entities defined with the JSM database.

Note that no attempt is made by JSM to check for the length and/or 'strength' of the password provided.

The application server's ini file option **[JadeSystemsManager] OperatorAliasPrefix** may be used to prefix the user id above before Windows or internal JSM authentication is attempted.

Example. If the OperatorAliasPrefix is LMH and the user id entered above is myname then the actual derived user id is LMHmyname and this is what is used for authentication.

The application server's ini file option **[JadeSystemsManager] WindowAuthenticateOnly** may be used to authenticate against the Windows domain only and does not attempt to authenticate against the internal JSM operator entity. This logon type uses the user id entered in the logon form, regardless of the **OperatorAliasPrefix** setting. This means that the length and/or 'strength' of supplied passwords or the aging of passwords as defined by domain policy settings is enforced.

Hot Keys

Several menu items are used repeatedly and have been assigned the following hot key values:

F2 - System View
F3 - Alarms Monitor
F4 - Backup Schedule

F5 - Unclaimed Alerts
F6 - JADE Environment Maintenance
F7 - List JARI's
CTRL+D – View SLA Documentation

Tool Bar Icons

Many of the most used JSM functions have icons on the tool bar which can be used instead of menu items

System View

Environment Version Status

Deployment Control

Deployment Summary

Backup Schedule

Monitoring Processes

Monitor System Stats

Alerts for <logged on user>

Unclaimed Alerts

Alert Set Maintenance

Alert Definition Maintenance

Toggle Navigation Panel visibility

Production vs Test

With users having connection to production and UAT copies of the JADECare Systems Manager, being able to differentiate between these on the desktop's task bar and on the various forms is desirable to avoid updating the wrong system. The ability to set a test mode option has been added to the initialization file and to allow the icon appearing on the desktop's task bar to be changed.

If the initialization file section **[Karma]** contains a keyword **TestMode** and the value is greater than zero, then the characters **[TEST]** will be inserted into the MDI frame caption.

If the above option is set and an icon file titled **JSM_Test.ico** is resident in the database server's server/c_misc directory, then this icon will be displayed on the task bar rather than the default JSM icon.

Setting Up For JADECare Environment Management

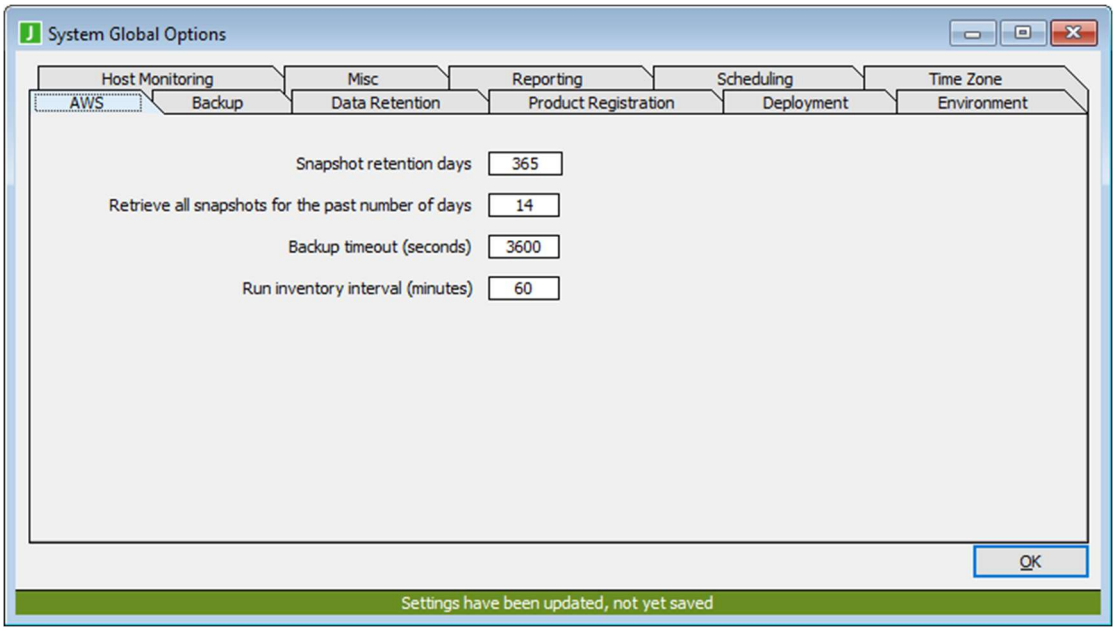
Introduction

This chapter describes administration functions required to set up the JADECare Systems Manager

System Global Options

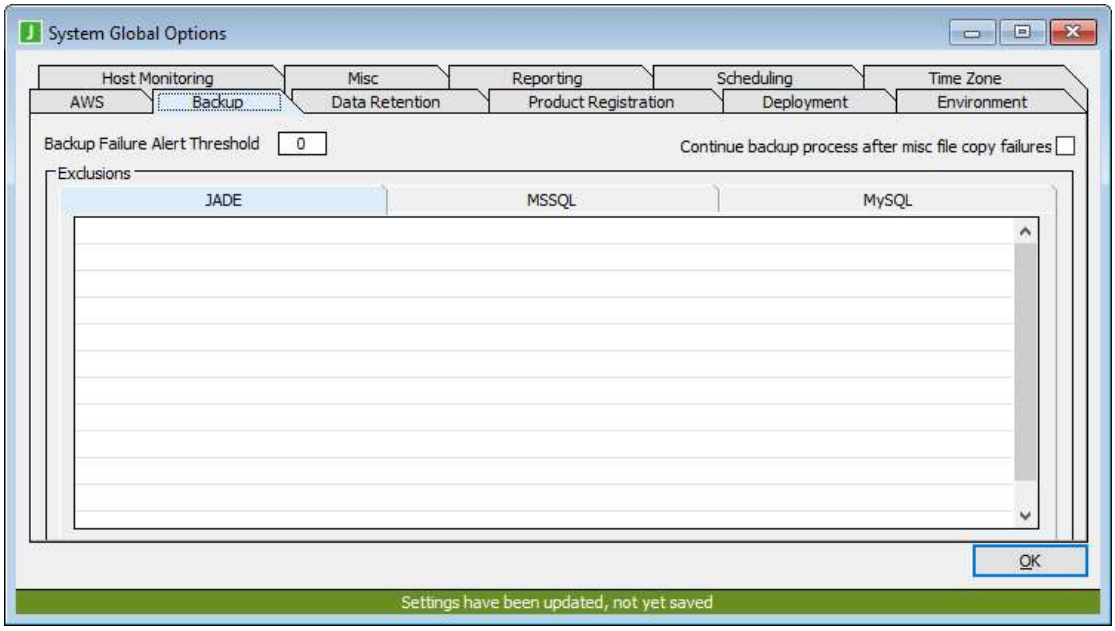
The system global options form now has several sheets covering different areas required to be setup. A copy of each of these sheets is shown below:

AWS Sheet



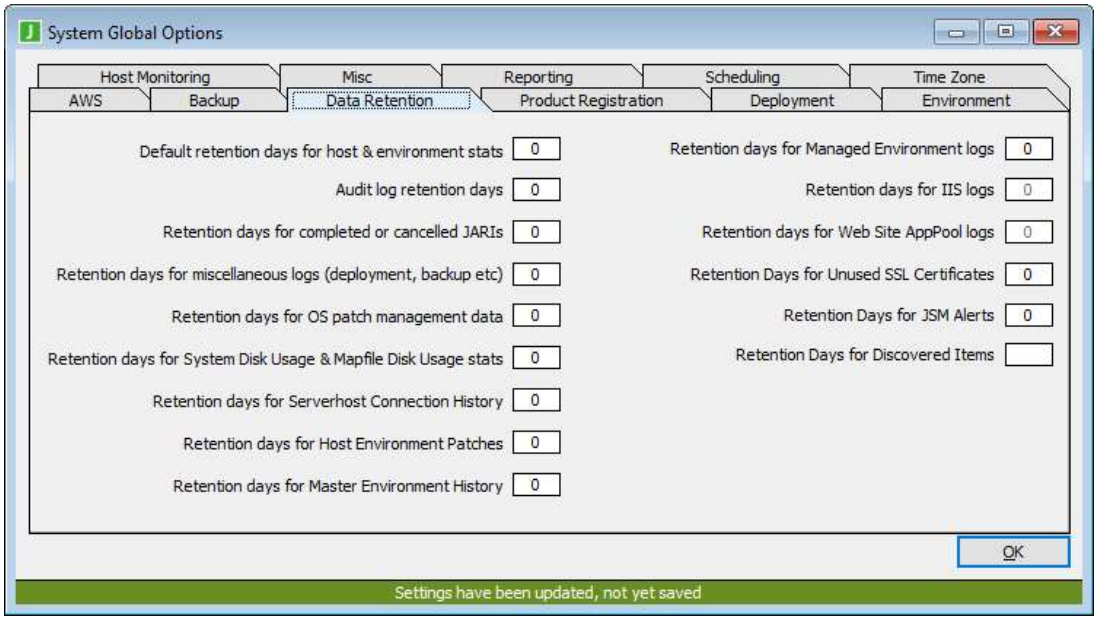
Snapshot retention days	Default value for snapshot retention days when no value is specified.
Retrieve all snapshots for the past number of days	Limit the number of existing snapshots retrieved from AWS when performing the inventory process.
Backup timeout	Backup timeout before giving up.
Run inventory interval (minutes)	Conducts an inventory of AWS snapshots every <number of minutes> specified.

Backup Sheet



Backup Failure Alert Threshold	Allows users to define the number of successive backup failures that may occur before JSM will generate a separate alert that this many successive backups have failed (Multiple successive backup failures for <env>)
Continue backup process after misc file copy failures	Setting this option allows the backup to continue to completion even if files under the <emv>/server/c_misc directory did not successfully copy. Regardless of this setting an alert will be generated advising of the files where the copy failed.
Backup Exclusions	This identifies directories, files or directory/file masks that are to be excluded globally across all JSM Managed Backups differentiated by the database type (JADE, MSSQL). The name entered is automatically prefixed with the drive letter / mount point and the managed environment root name.

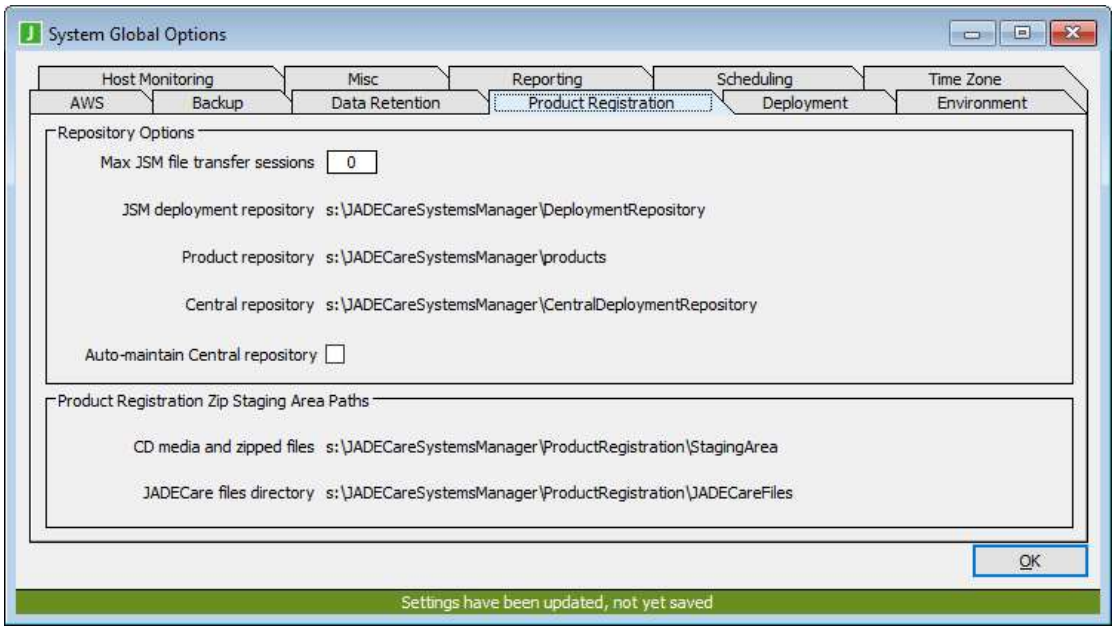
Data Retention



Default retention days for host & environment stats	This is the value that new server host and new host environments will default to when they are created.
Audit log retention days	Period of time (days) for which audit log details, as viewed via menu **View | Audit Log**, are retained for.
Retention days for completed or cancelled JARIs	Period of time (days) for which JARIs which are completed or cancelled and do not have the **Retain** option specified are to be held for.
Retention days for miscellaneous logs (deployment, backup etc)	Period of time (days) for which summary logs (backup, compaction, installation, restores, etc) are to be held for.
Retention days for OS patch management data	Period of time (days) for which O./S patch data is to be held for.
Retention days for System Disk Usage & Mapfile Disk Usage stats	Period of time (days) for which System Disk statistics and Mapfile Disk Usage statistics are to be held for.
Retention days for Serverhost Connection History	Period of time (days) for which to retain the connection history for each serverhost
Retention days for Host Environment Patches	Period of time (days) for which to retain the host environment patch history (schema, ddbfile, methods, classes etc)
Retention days for Master Environment History	Period of time (days) for which to retain the master environment history.

Retention days for Managed Environment logs	Inherited by following when entities defined 'log days' is zero: - JADE Environments - ODBC Environments - .Net Services
Retention days for IIS logs	Period of time (days) for which to retain Web Site logs. Inherited by websites when defined 'IIS Log Retention' period is zero.
Retention days for Web Site AppPool logs	Period of time (days) for which to retain Web AppPool logs. Inherited when 'log days' for web apppool is zero.
Retention Days for Unused SSL Certificates	Period of time (days) for which to retain SSL certificates. For JADE App Server certificates this is the date last used.
Retention Days for Discovered Items	Period of time (days) for which to retain discovered items (such as web sites and certificates). The period is relative to the entity logical deletion date.

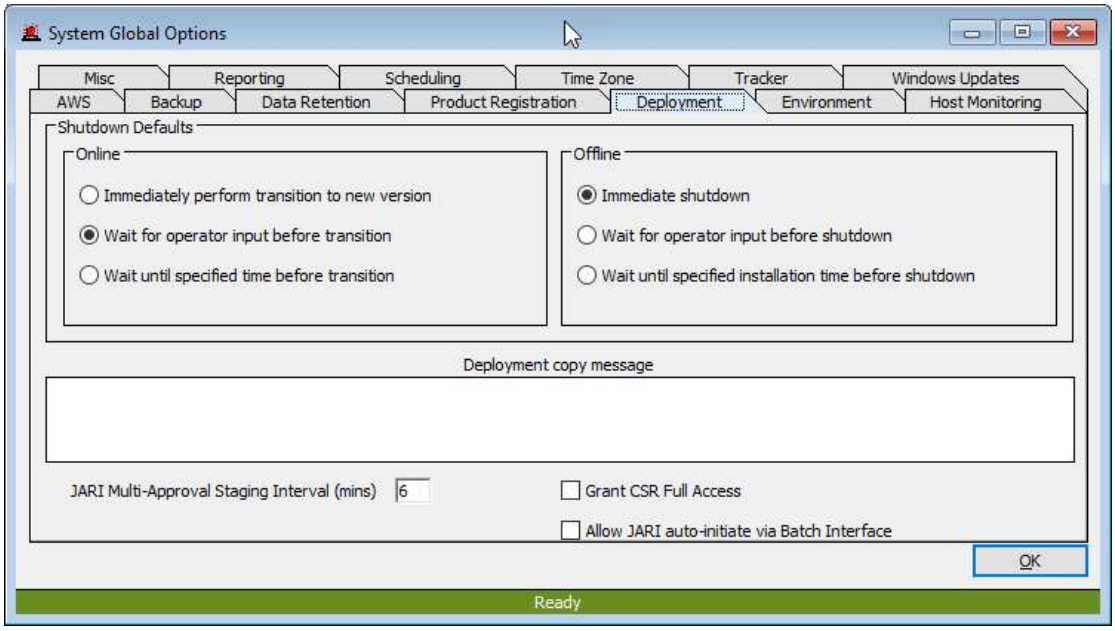
Product Registration



Auto-maintain Central repository	Setting this option will: - initiate application **JossSetupCentralDeployRepos** to define the appropriate sub-directories for every master environment defined in JSM and declare appropriate permissions for every developer with access to each master environment.

	- Cause any changes to master environments to be reflected in the Central Repository structures - Cause any changes in access to a master environment by development staff to be automatically maintained as permissions in the Central Repository structures.
Central repository	This shows the pre-deployment repository where a developer may place files prior to starting the JARI build. The intent here is to avoid having to send files to the local PC of the person doing the JARI build and then pulling these files from that PC back to the JSM host (i.e. NZ to UK back to NZ- see section “Optional Deployment File Source” for further details). The files pre-deployed here will be included when the check box ‘Include file from Central Deployment Repository’ on the JARI build form is ticked.
Deployment Copy Message	Define the default deployment copy message which is to be used as the default if no specific message is entered in the deployment form.
JSM deployment repository	Shows the work area where deployment files are to be staged when building a JADE Managed System release.
Max JSM file transfer sessions	Defines the number of concurrent deployment distributions that may be active at any one time.
Product repository	Shows the directory definition from where CardSchema and JADE releases (including hot fixes) are to be obtained from when requested in a JARI.
CD media and zipped files	Shows the directory definition where new JADE releases (from CD media) / CardSchema releases and ParSys downloaded hotfixes are located for the product registration process.
JADE Care files directory	The directory definition where mainly static files provided by JADECare Tool Sets are held for the product registration process. This includes scripts, cnwplugin.dll etc.

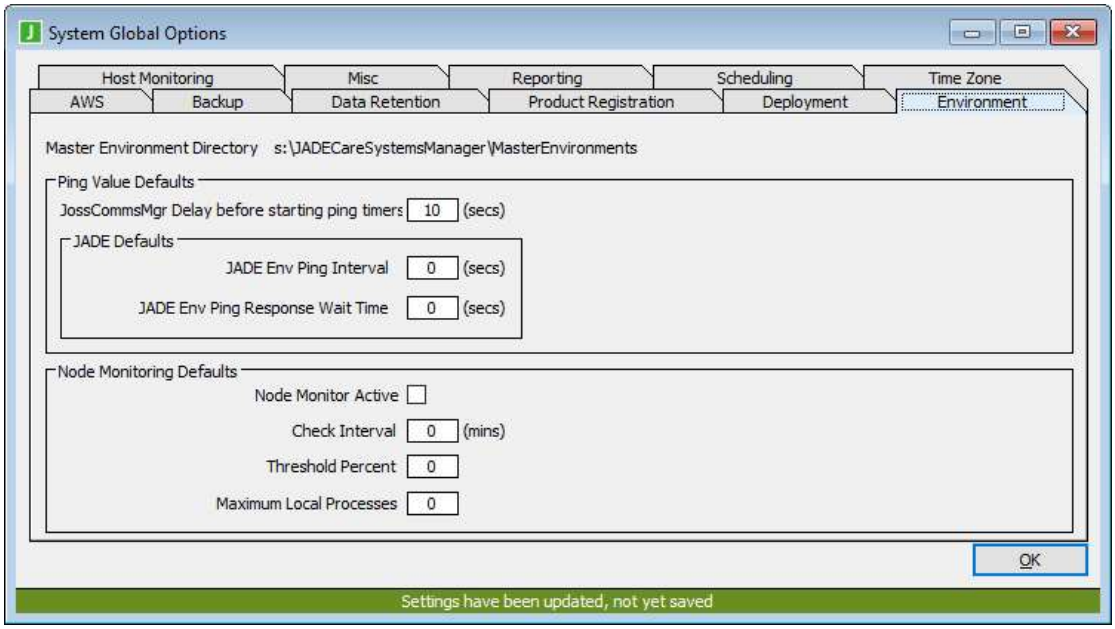
Deployment



Allows :

- site specific Shut Down options to be specified for online / offline deployments
- definition of a Deployment Copy message
- Fore JADECare 24 only, allows the definition of the staging interval for the JARI multi-approval feature.
- Allow or disallow the JARI auto-initiate command line feature

Environment



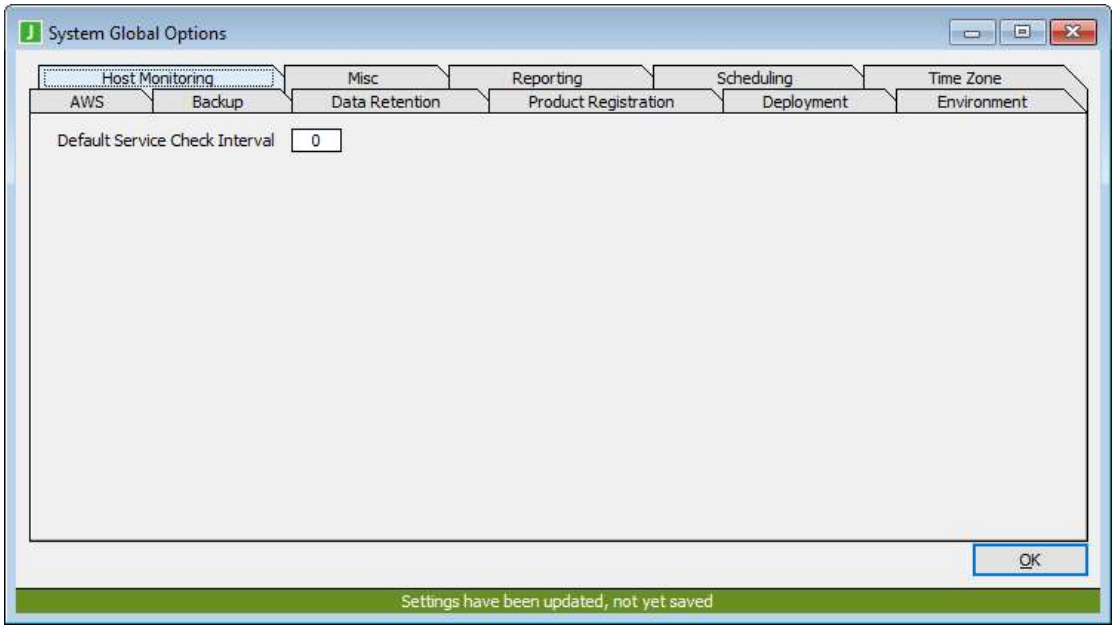
Master Environment Directory	The directory where all miscellaneous files reside for each Master Environment (version files, thinclient files, some specific deployed files.

JossCommsMgr Delay before starting ping timers	the number of seconds to wait after start of JossCommsMgr before activating pinging of JADE managed systems (where pinging has been requested
JADE Env Ping Interval	when a new JADE Managed System is defined (either by 'discovery' or via JADE Environment screen) the value provided here is used to define the JADE Managed System's Ping Interval If the JADE Managed System's ping interval is reset to zero, the next time the system changes to active it will set its ping interval to this default value
JADE Env Ping Response Wait Time	When a new JADE Managed System is defined, this value is used to define the wait interval for the ping response before an alert is generated. If the JADE Managed System's ping interval is reset to zero, the next time the system changes to active it will set its wait for ping response interval to this default value.
JOOB Env Ping Interval	when a new JOOB Managed System is defined the value provided here is used to define the JOOB Managed System's Ping Interval If the JOOB Managed System's ping interval is reset to zero, the next time the system changes to active it will set its ping interval to this default value
JOOB Env Ping Response Wait Time	When a new JOOB Managed System is defined, this value is used to define the wait interval for the ping response before an alert is generated. If the JOOB Managed System's ping interval is reset to zero, the next time the system changes to active it will set its wait for ping response interval to this default value.
Node Monitor Active	This turns node monitoring on or off for all JADE environments in JSM.

The following items are inherited as new JADE Environment entities are created.

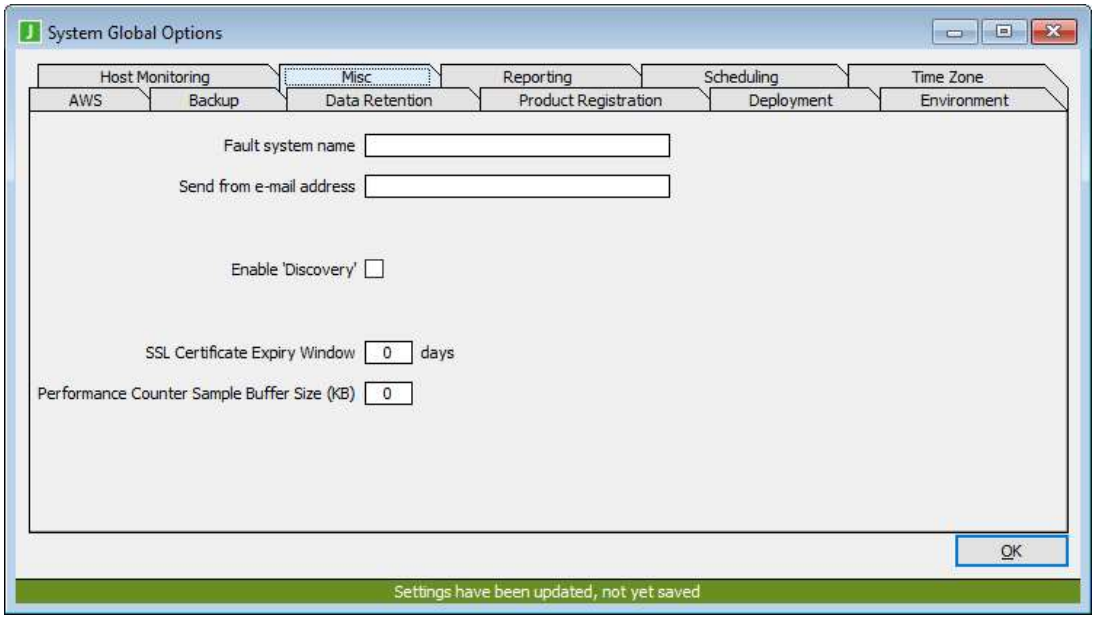
Check Interval	The default check interval value - inherited by any newly created host environments.
Threshold Percent	The default threshold percent value - inherited by any newly created host environments.
Maximum Local Processes	The default number of local processes value - inherited by any newly created host environments.

Host Monitoring



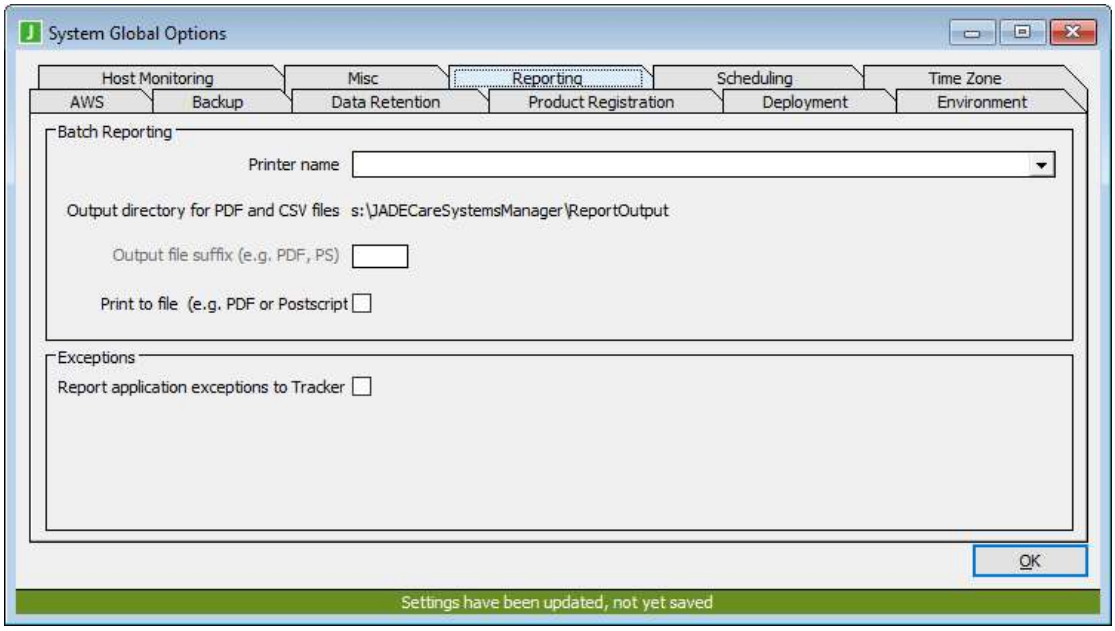
Default Service Check Interval	defines default timer interval to be used as a default when setting up new hosts and services on those hosts. If server host is defined with a zero service check interval, the default value specified here is used.

Misc



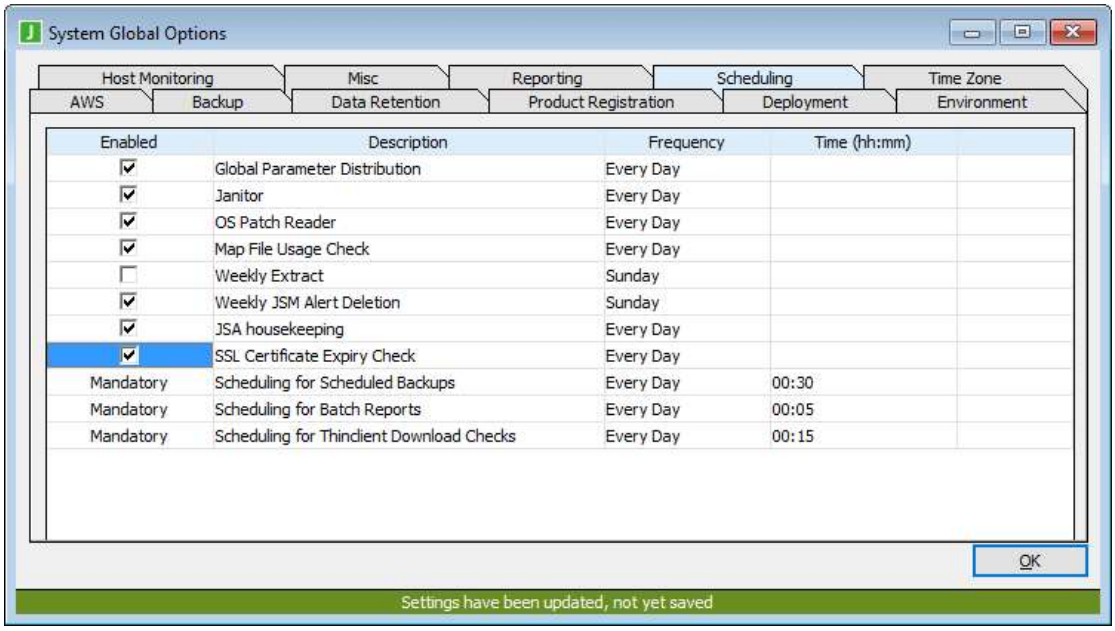
Fault system name	Identifies name of local fault system where one is used. In the case of JADE Software Corporation this is called JADECare Tracker and is a module using the JADECare Systems Manager database.
Send from e-mail address	The email address to be used as the sender and the reply to where automated messages are sent by the JADECare applications (eg deployment and Tracker system)
Disaster recovery directory	Shows the value of the ini file setting used for disaster recovery data extracts. Only used by JADE Software Corporation.
Enable Discovery	Global control to enable/disable .Net service and Web site discovery
SSL Certificate Expiry Window	Period of time (in days) which will be used to check and raise an advisory for any SSL certificate is about to expire (value must be > 0).
Performance Counter Sample Buffer Size (KB)	Defines the default performane counter buffer size to be used by JSA when accessing Windows Performance Counters. This value is inherited by each server unless a specific value has been defined for the server.

Reporting



Printer name	The printer name. This is selectable from the printers visible on the JSM database host.
Output directory for PDF and CSV files	Shows the ini file value used where any PDF and / or CSV files generated by batch reporting will to be directed. Reports generated are placed into a sub-directory qualified by customer id, the year and month and year.
Output file suffix	May specify PDF or PS.
Print to File	Check box. Set to true if PDF or PostScript output is always required for batch reports.
Report application exceptions to Tracker	This may only be set when the Tracker module is implemented. All unhandled JADE exceptions from applications will automatically create Tracker faults. These faults default to priority B, the priority and whether an alert is still raised may be specified by Host Environment application.

Scheduling



The above sheet identifies the JSM and some JSA specific items that may be (enabled / disabled) or must be (mandatory) automatically scheduled. The times at which these events execute may be modified using the above sheet.

Notes Regarding Daylight Saving Changeovers

Events scheduled to execute during spring forward will not execute and an alert will be generated. For example, on 27 September 2015 as daylight saving is entered in New Zealand, New Zealand clocks move forward one hour; 2am becomes 3am. An event scheduled for 2:30am will not execute on 27 Sept 2015. Two strategies for dealing with this are:
1) Add a one-off schedule an hour later on the DST switchover day. E.g. add a schedule to execute at 3:30am.
2) Avoid scheduling activities during the spring forward hour in the time zone where the schedule executes. This strategy avoids the issue of dealing with an alert generated for the skipped schedule.

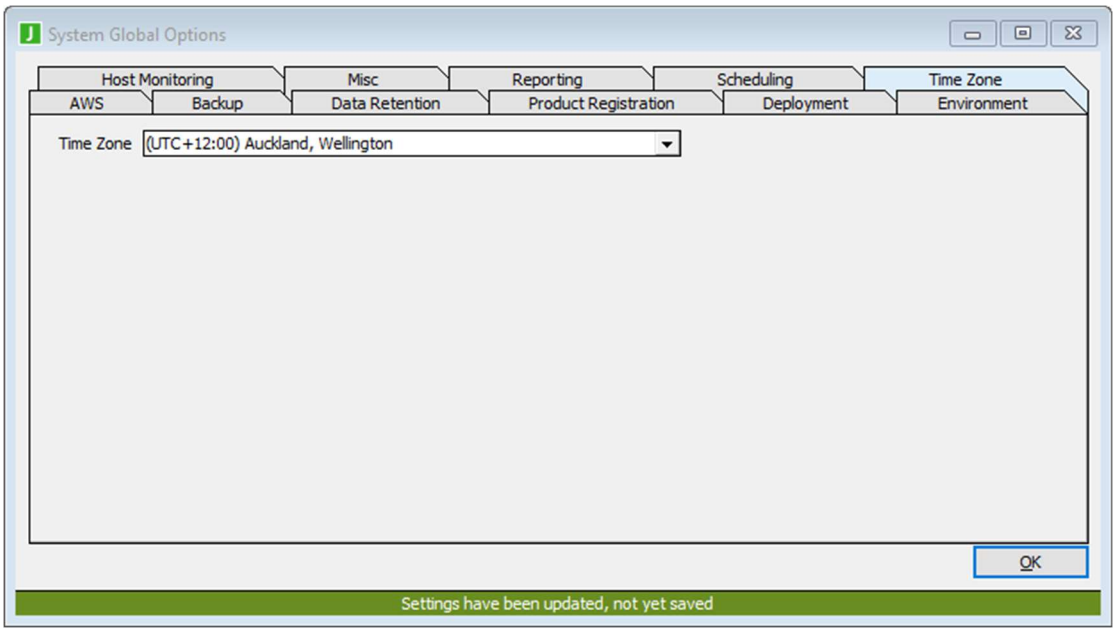
Events scheduled during the fall back hour (3am becomes 2am in New Zealand) will execute normally (only once even though the hour 2am-3am is repeated). No special consideration is required.

Global Param Distribution	This is an optional item and is applicable to JSM only. This causes the global parameter file distribution to occur on a daily basis
Janitor	This is an optional item and is applicable to JSM only. This causes the deletion process to execute and delete old statistics, audit entries. If Janitor is enabled and 'Unused/logically deleted Certificate retention days' value in 'Data Retention' greater than 0, process will check and delete any expired SSL certificates (expired Jade application server or logically deleted Web site) which does not fall in to retention period.

Weekly Extract	This is an optional item and is applicable to JSM only. This causes: - the extraction of access profiles for each user on a weekly basis. - if Tracker instantiated and DR directory specified, extracts all SLA contact details to file All_Customer_Contact_Details.htm.
JSA Housekeeping	This is an optional item and applies to JSA only. Files older than that defined by the log days value for the JSA environment are cleaned up (or those older than seven days if this value is not set). Old journals are removed from the current and archive database directories. Temp files are removed from the transient database directory. Files with extensions tmp, log, csv, zip, rar, dmp, bat, end, err and txt are removed from the logs, cardlog, activity log, crashlogs and processdumps directories.
SSL Certificate Expiry Check	This is an optional item. When enabled checks will occur on certificate expiry using the period in 'SSL Certificate Expiry Window' in 'Misc' tab. If a certificate is found to be in the expiry window then one of the following advisory mechanisms are used: • If Tracker is instantiated then a call will be generated and assigned to appropriate FMS assignee: o for IIS certificate this will be the server's FMS assignee o for JADE application server this will be the JADE environment's FMS assignee. o If FMS assignee is not specified or invalid at the time the default assignee will be that declared as Operations Role Id in ini file. • If Tracker is not instantiated then a level 7 JSM alert will be raised defining either the web bindings and websites or the JADE environment.
Scheduling for Scheduled Backups	Mandatory item and applies to both JSM and JSA. This process is used to schedule the coming days scheduled backups as specified for each host environment.

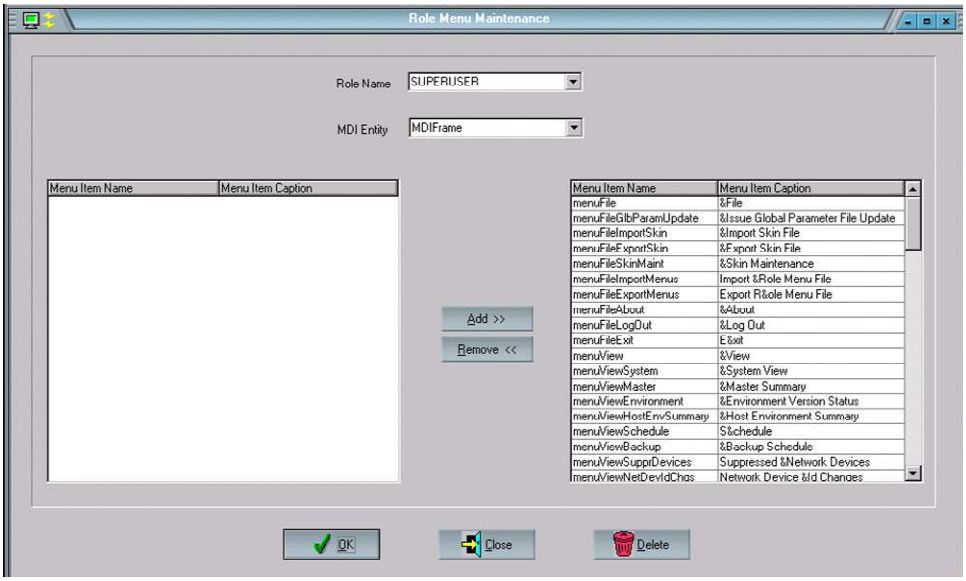
Scheduling for Batch Reports	Mandatory item and applicable to JSM only. This process is used to schedule the coming days report generation as specified in the report set setup.
Scheduling of Tracker reminders	Only seen if Tracker module implemented. Mandatory item and applicable to JSM only. This process is used to schedule Tracker reminders for the coming day..
OS Patch Reader	Scheduled time for the OS Patch Reader application to process the ServerHost Microsoft hot fix flat file information.
Map File Disk Usage Check	Scheduled time held in JSA to check each of the ServerHost held Host Environments map file sizes for reporting purposes.

Time Zone



Time Zone	Defines the current timezone this JSM installation resides in.

Role Security Setup



The JADECare Systems Manager database has more than one schema view into it and there will eventually be several MDI entities each having its own menu structure. The menu items that are visible and may be selected is allocated by the Role or Roles that a person has when logging into JSM or one of its views.

The above screen provides JSM Developers to allocate visibility of menu items to each of the roles currently available.

Roles currently defined and their background are:

Admin
The highest security classification for a client. Typically a user allocated this role may update **Operator** details and other customer specific information for entities associated with his/her customer.

CSR
A customer services representative role where the person concerned looks after the day-to-day relationship between the provider of the JADECare Systems Managed Service and the client whose JADE systems are being managed. This role is expected to authorise changes being made to JADE managed systems via the JARI process.

Default
the lowest security permitted to access JSM or any of its views.

Developer
A person that does development work for a JADE Managed System and may submit a JARI.

ITManager
Only applicable to customer staff. This role is the client equivalent of 'CSR' however restricted to only the customer's environments and services.

JadeStaff
Deemed to be staff internal to the company providing the JADECare Systems Managed Service and which is executing the JADECare Systems Manager.

Operator
Staff who man the Help Desk and monitor JADE Managed Systems. They have a subset of the features used by SuperUsers.

SuperUser
Staff who install, maintain, monitor and fix issues with JADE Managed Systems and require access to all the features provided by JSM

The menu items displayed are derived from accessing the metaschema for the MDI entities named. Only two MDI entities currently are declared:

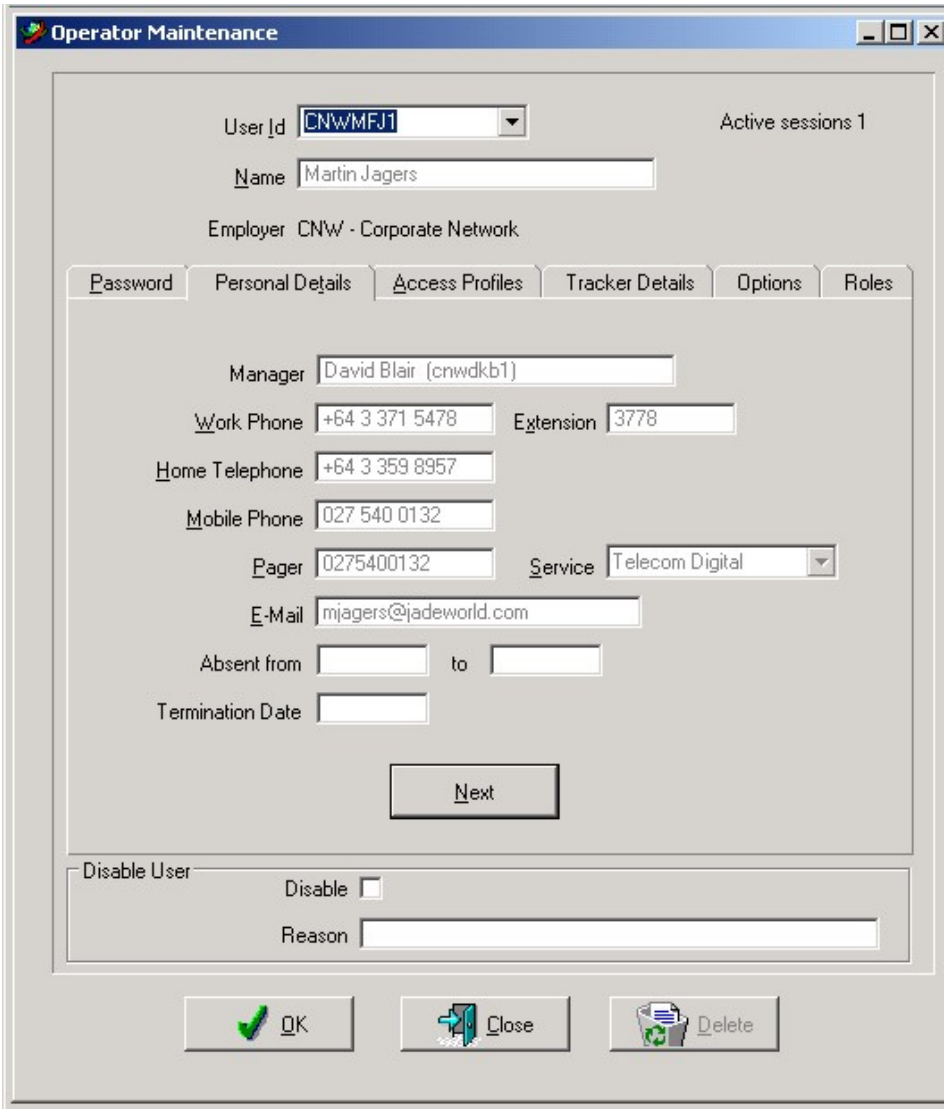
MDIFrame
The form that is named as J.S.M's application MDI Frame and which inherits all the menu items from its parent(s).

ScmMDIFrame
This form is part of the Tracker module which is released as a separately priced module.

The Role Security above will have been pre-defined and a file containing these will be provided as part of the JSM 2.7 release. This file can be loaded via the menu **File | Import Role Menu File**. The file is to be placed in JSM's \server\c_misc directory on the database server host and is expected to be called 'jsm_menu_extract.txt'.

SecurityMDIFrame
This is module yet to be implemented.

JSM User Setup



NOTE : The sheets Tracker Details and Roles shown on the above form are only visible if the JADECare Tracker module has been implemented.

Users are defined via the menu selection **Set Up | Operators**. The above screen is shown and displayed details of the user currently logged on.

Existing user details can be viewed by anyone and is dictated by the role(s) allocated and the customers defined for those roles. The extent of update capability and what may be viewed is defined by the highest security role defined by your logon user-id.

An existing user may be declared as ‘Disabled’ by clicking on the checkbox shown and a reason must then be provided. This causes the log-on to be rejected and produces a message box stating ‘User-id is disabled’.

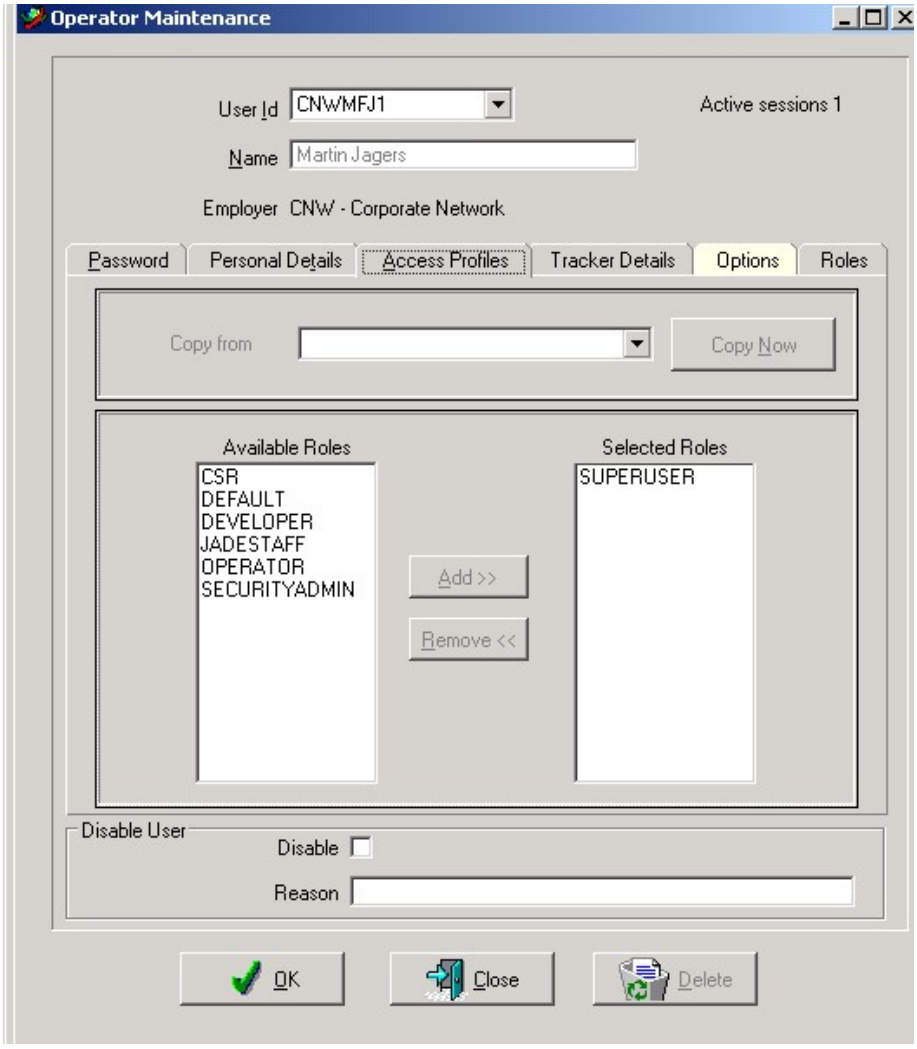
A new user can be defined by entering a unique identifier into the field specified ‘User Id’. Note that the first 3 characters of this user-id must be the same as an existing customer id.

The full name of the user can be entered in the ‘Name’ field and the employer needs to be selected from the list of available customers..

If defining a new user, then the password will be required as the user will not be committed otherwise. This will create the skeleton user definition and open the 'Personal Details' folder.

If LDAP is in-use and the employer of this user has been defined in LDAP, then all users for this employer are to be defined in LDAP. Whenever such a user is defined, LDAP is referenced to refresh all the personal details and may not be overwritten.

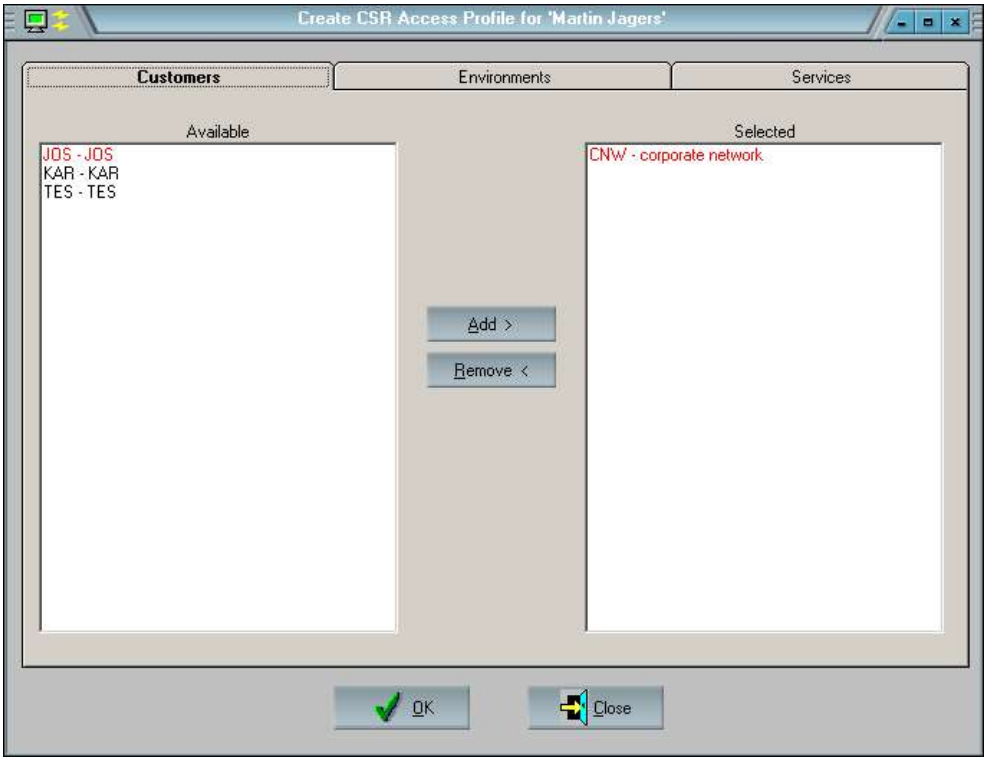
Once the 'Personal Details' have been defined, and if a new user and you are logged on as a super user, then the third folder is opened 'Access Profiles'.



This folder allows the definition of one or more security roles to a user. These security roles allow or disallow access to various menu items and may also restrict visibility to information. Refer to **Role Security** for details and definitions of **Role**.

If no roles have yet been defined, then another user may be selected and all access profiles for all security roles for nominated user may be copied (Copy from combo box and Copy Now button will be enabled). Once one or more security roles have been copied, double clicking on the security role in the Selected Roles control will allow the access profile to be viewed and edited.

By selecting an item in Available Roles and clicking add (or double-clicking on the item in Available Roles) a modal form is presented as follows:



This screen is used to allocate the customers that a user may access within this role. Once the customers have been defined, then the environments for those customers may be allocated if the customer has 'Explicit Security' set (these are highlighted in red, refer to **Customer Setup** below).

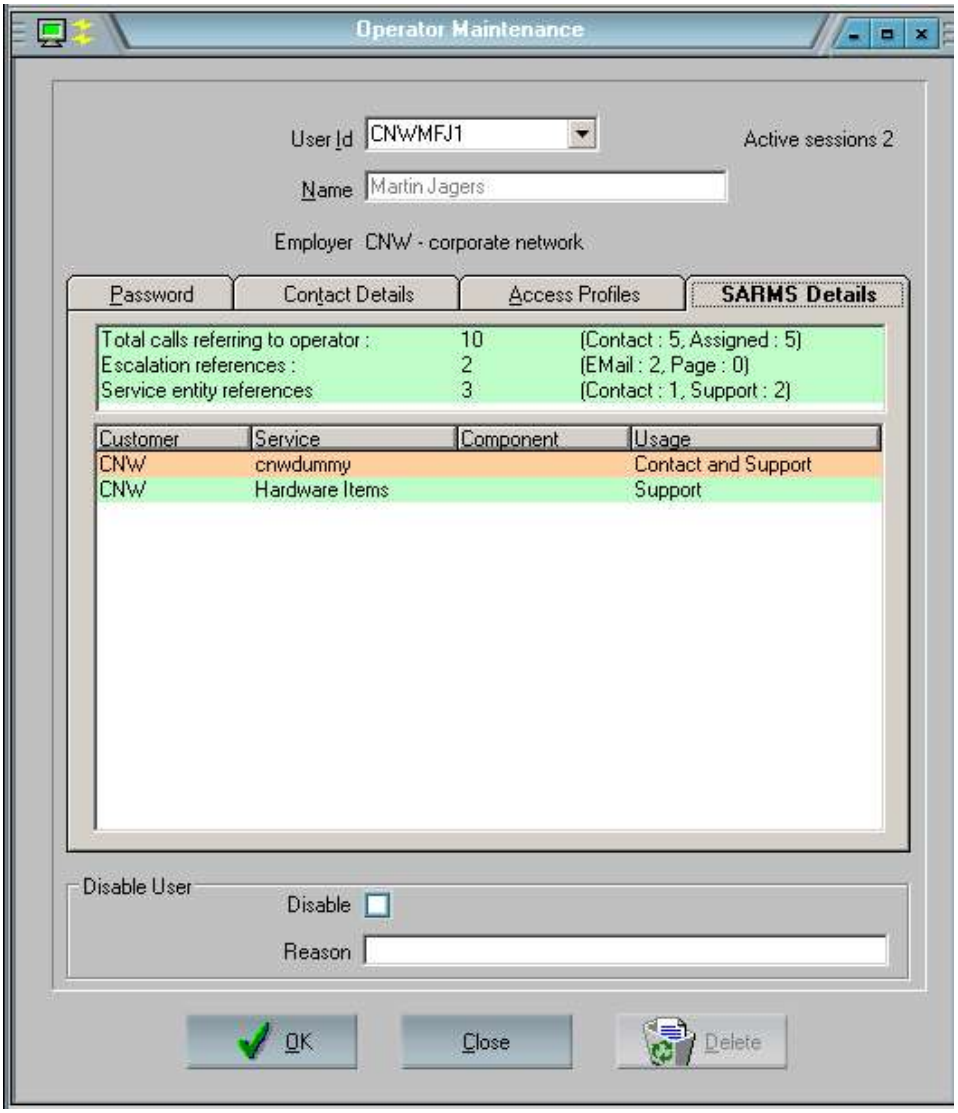
By default, the customer (employer) of the user is already selected and shown in the 'Selected' column. Any other customers displayed can be selected. Customers can be de-selected by selection and click 'Remove' or double-clinking on the entity in the 'Selected' column

Clicking on the Environments tab will open that folder. The environments listed here are dependent on the customers selected. Those customers that do not have 'Explicit Security' set will show all their environments in light grey in the 'Selected' column. Those customers with 'Explicit Security' set, need to have their environments explicitly selected before details of those environments can be viewed by the user for who this role is being set up.

To de-select an environment (which is not light-grey), either select and click Remove or double-click on the entity in the 'Selected' column.

When selections have been made, click on OK to save these Role specific details for this user.

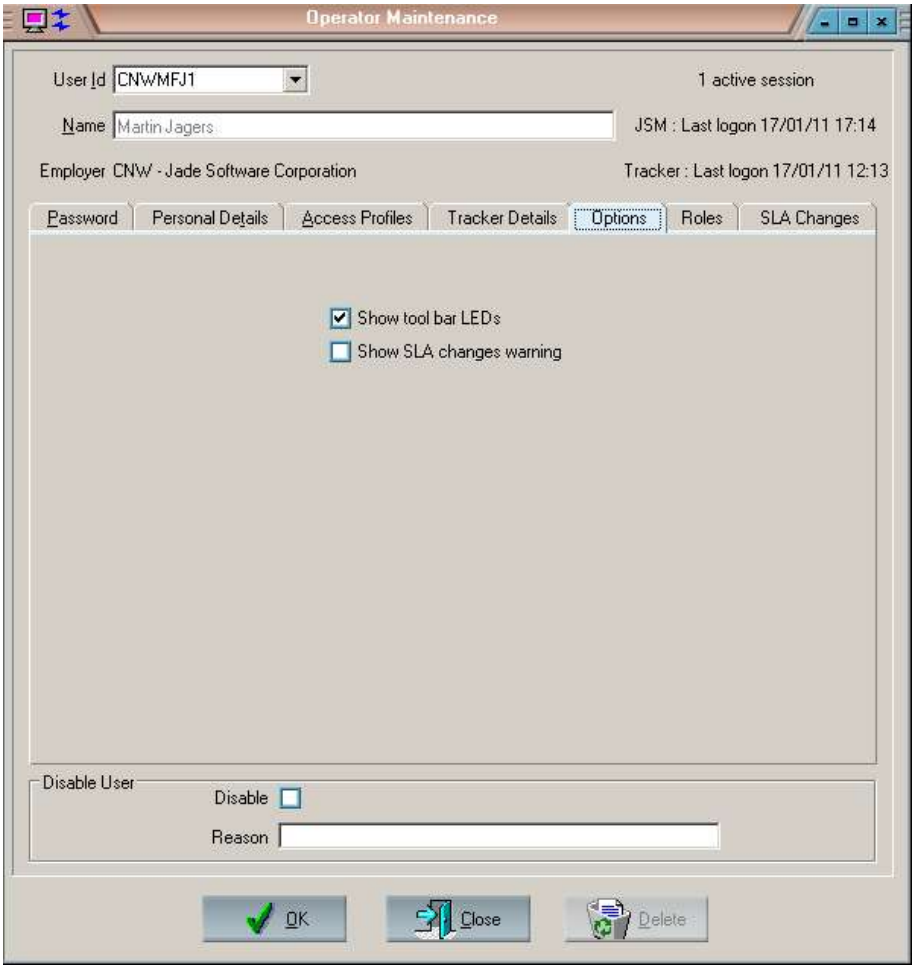
When an environment is marked as selected or de-selected, any corresponding Tracker service is also selected / de-selected. Note however, that if a Tracker service is selected / de-selected, the corresponding environment (if there is one) is NOT automatically selected / de-selected.



The Tracker Details tab is only visible if the JADECare Tracker module has been implemented.

The first table provides details of any references to this user. Where one or more references are shown to exist, this would preclude the user from being deleted.

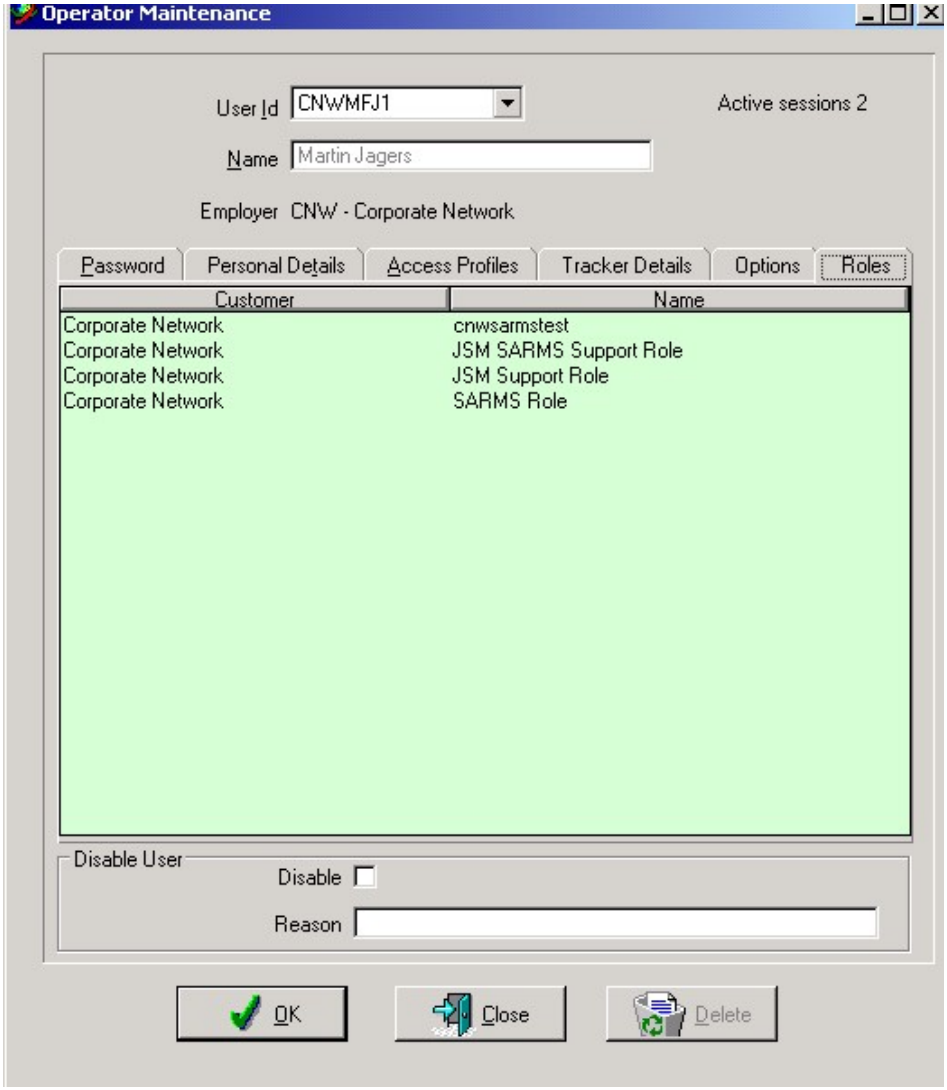
The second table provides details as to which service and component entities this user is being referenced by and the type of reference (contact, support or both).



The options sheet allows the operator to enable / disable start up options:

- the LED display showing JSM heartbeat (all is well)
- for staff defined as SLA Administrators or Updaters, whether or not to show whether any SLA items have been changed.

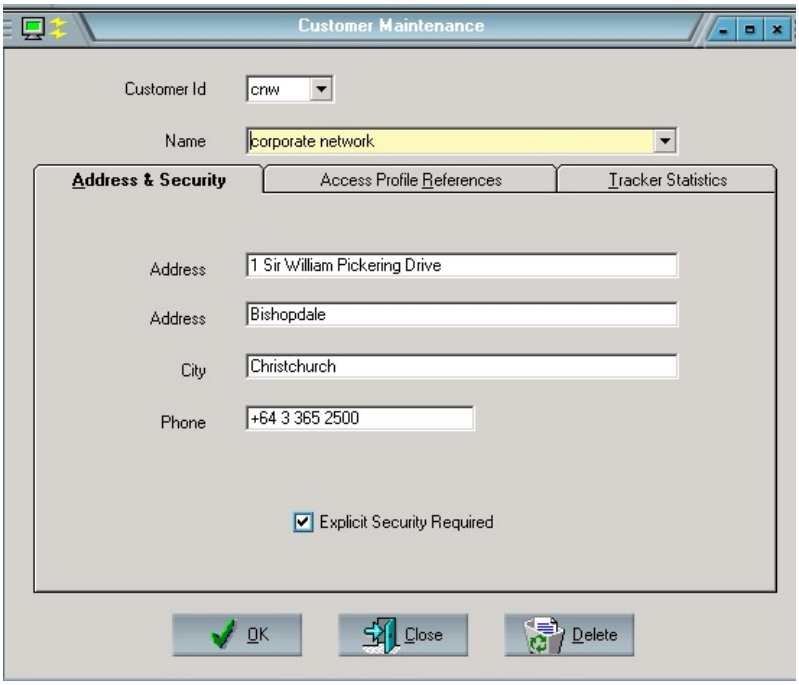
These values are retained across sessions.



The Roles sheet is only visible if the JADECare Tracker module has been implemented. This sheet identifies the roles which this user is an occupant of.

Where the customer has been defined in LDAP (and LDAP is in use) the information above is refreshed from LDAP whenever this user signs on or is the target of any emails due to reminders and/or escalation and/or call assignment.

Customer Definition



Customers can be defined by this form. To view existing customers, the selection can be done either on the unique 3-character customer prefix or by the full name of the customer.

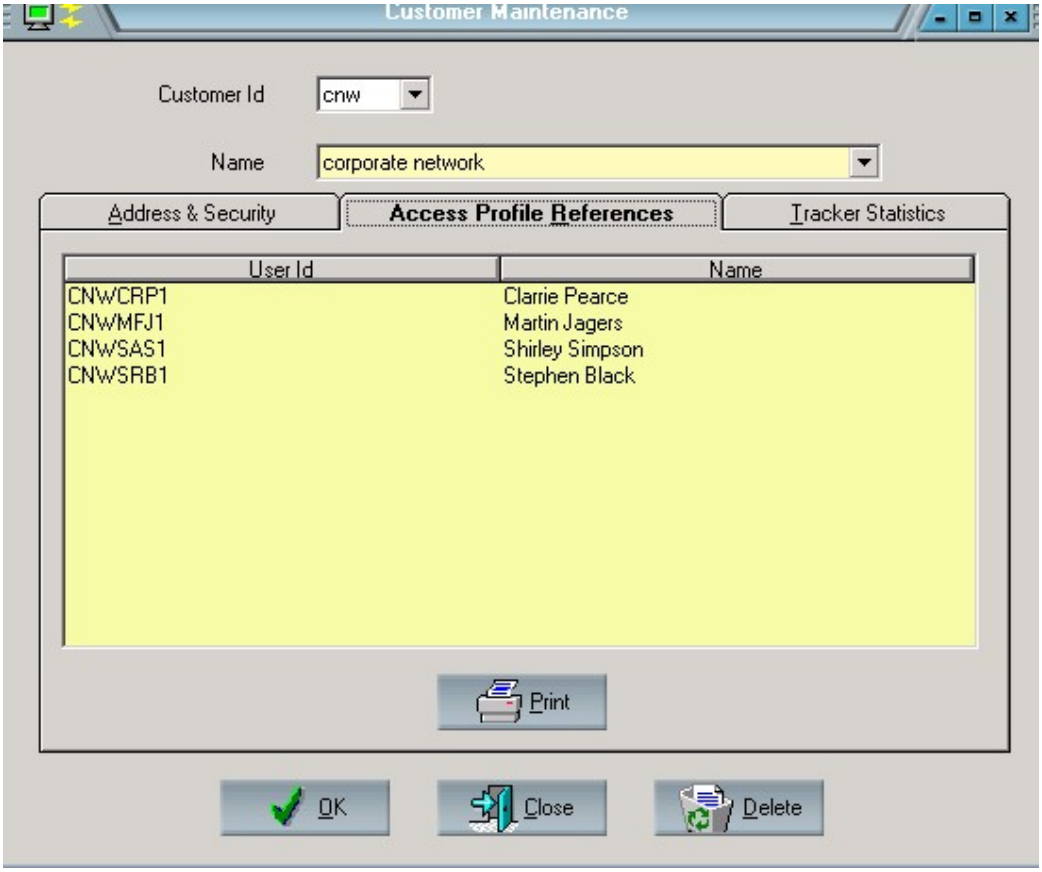
The name must be a unique item and may not be empty.

The address fields (address and city) and phone are free format fields and are optional.

The check box ‘Explicit Security’ identifies whether the environments for a customer need to be explicitly allocated to a user before the user can see details or if the user is implicitly allowed to view these details by being able to view the customer entity.

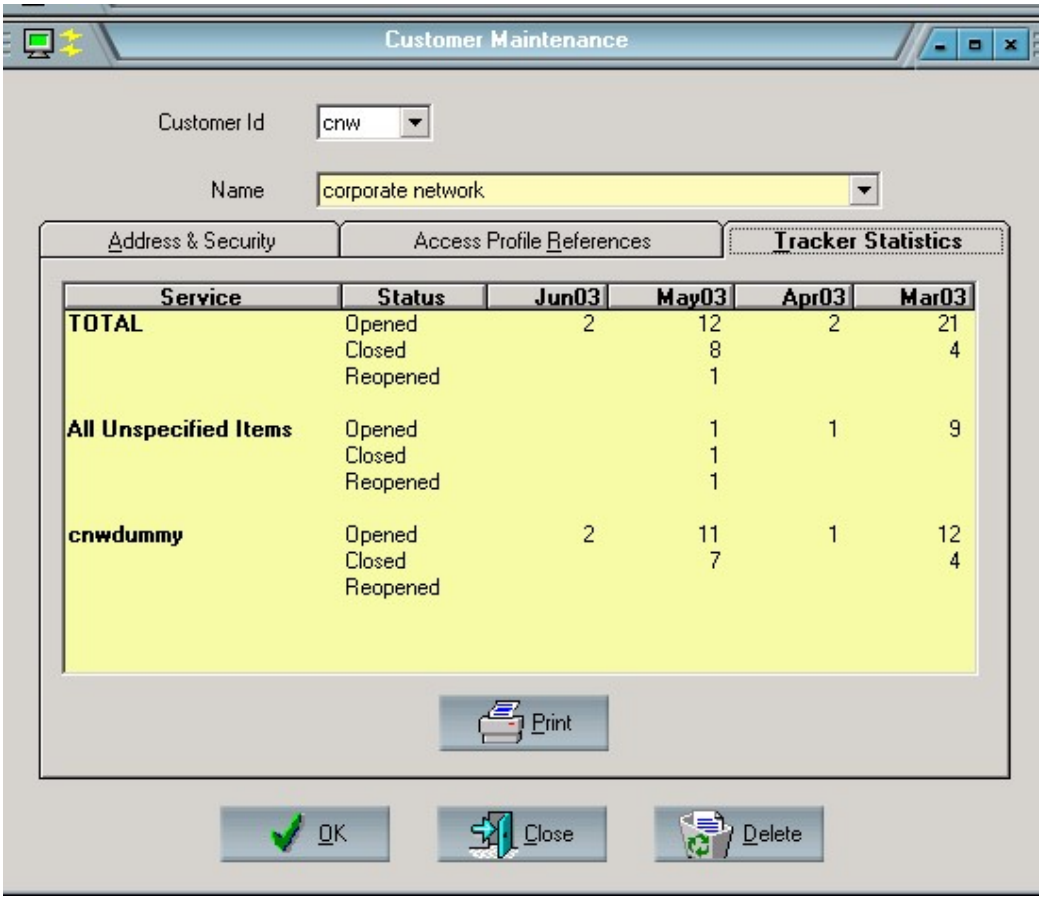
If the Tracker module has been implemented then two other sheets are provided otherwise just one additional sheet may be seen.

Access Profile References



The sheet shown above identifies which users have been granted access to this customer. The specific details of the type of access and which environments and services the users above may access can be viewed via the Operator Maintenance form, and the Access Profile Maintenance form accessible from that.

Tracker Statistics



This sheet is only visible if the JADECare Tracker module has been implemented. This provides a statistical analysis of calls for the customer by service and also overall totals for the customer.

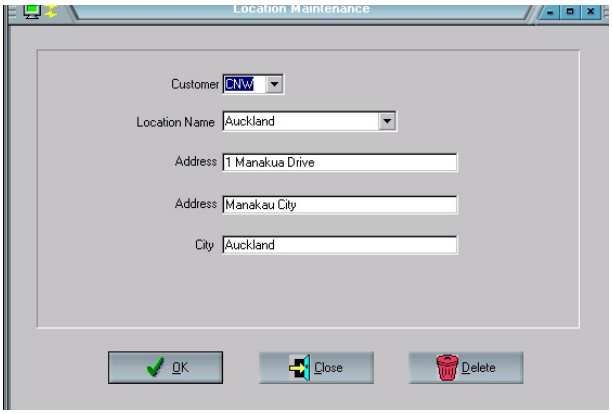
Opened — The count of calls opened in that month. This open count does NOT include the reopened count.

Closed — The count of calls closed in that month.

Reopened — The count of calls that were reopened in that month. This count is not a subset of the open count nor does it mean these calls are still open.

BEWARE. Calls opened against service A could be closed against service B and re-opened against service C.

Location Definition



One or more locations may be defined for a customer entity.

The location name must be unique within those locations already defined for the customer. The locations are used to provide meaningful information as to where a machine (server host) is located.

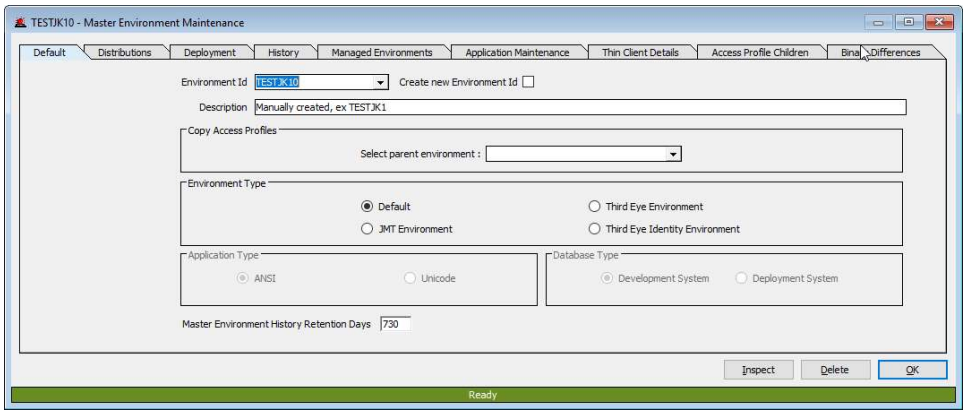
Environment Maintenance

JSM can be allowed to work by a process of discovery for JADE environments.

For JADE environments this occurs when a new JADE Managed System is installed, has the KarmaControl ini file setting set to true and instantiates a CnKarmaCntrl class of object when it is executed. This will cause JSA to detect its presence and forward details to JSM which will create the necessary environment details if they have not already been defined.

Environment Definition

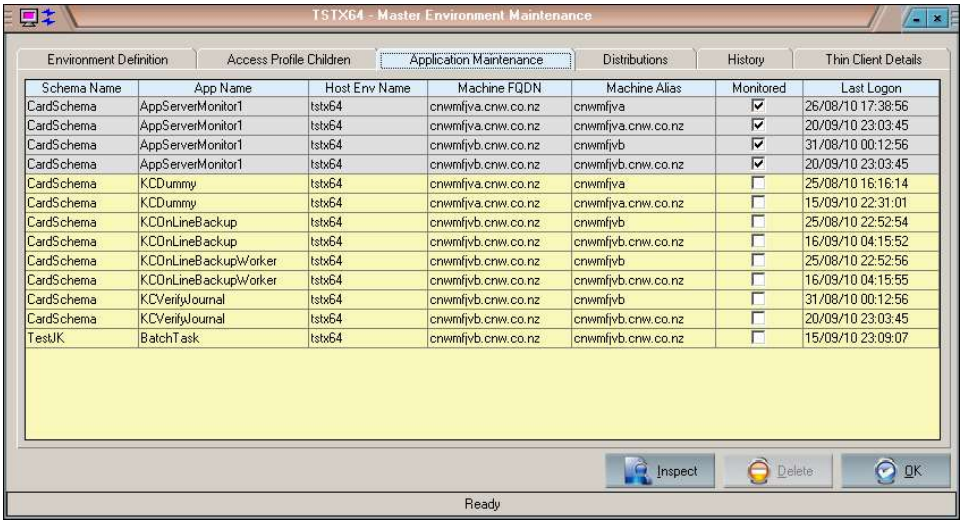
However, the environment may also be defined before the system is actually installed so that JSM is used to deploy the new system.



Create new Environment Id (check box)	Checking this allows the user to manually define a Master Environment
Description	If the master environment was created by the 'discovery' process, then the description is 'Auto-created'. Free-format field to describe the master host environment.
Environment Type	This control is only visible for JADECare 24 use. The option buttons allows the definition of all JADE Environments associated with this master environment to be any one of: • default • JMT Environment • Greentree Environment • ThirdEye • ThirdEye Identity Default – no specific handling required.a JMT Environment – allows specific handling for this environment type: • access to the JADE Master Terminal product • allows the use of the 'JMT Deployment' process.

	• Forces SDS Secondary to execute application version registration Greentree Environment – allows specific handling for this environment type, future. ThirdEye – allows specific handling for this environment type, future ThirdEye Identity – allows specific handling for this environment type, future.
Copy Access Profiles	Allows cloning an established access profile list from a parent environment to the selected Environment Id (child). Once this connection is established any entries added to the parent will be automatically carried over to the child. There are some pre-requisites: *The child must have an empty access profile list.* *The parent and child must have the some customer prefix.* *The parent must have explicit security set.* **Note:** Access profiles associated to a parent can be seen in the 'Access Profile Children' tab.
Application Type ANSI or Unicode option buttons	Identifies which application type applies to the master environment. Valid options are **ANSI** or **Unicode**. The default is **ANSI**.
Database Types: Development or Deployment option buttons	Identifies whether the master environment has the development suite of files and can be used for development or if deployment only
Set Up Directories… button	After displaying a caution form of what is about to be done, this will cause the necessary directories and version control files to be defined on the JSM server host for this master host environment.

Application Maintenance

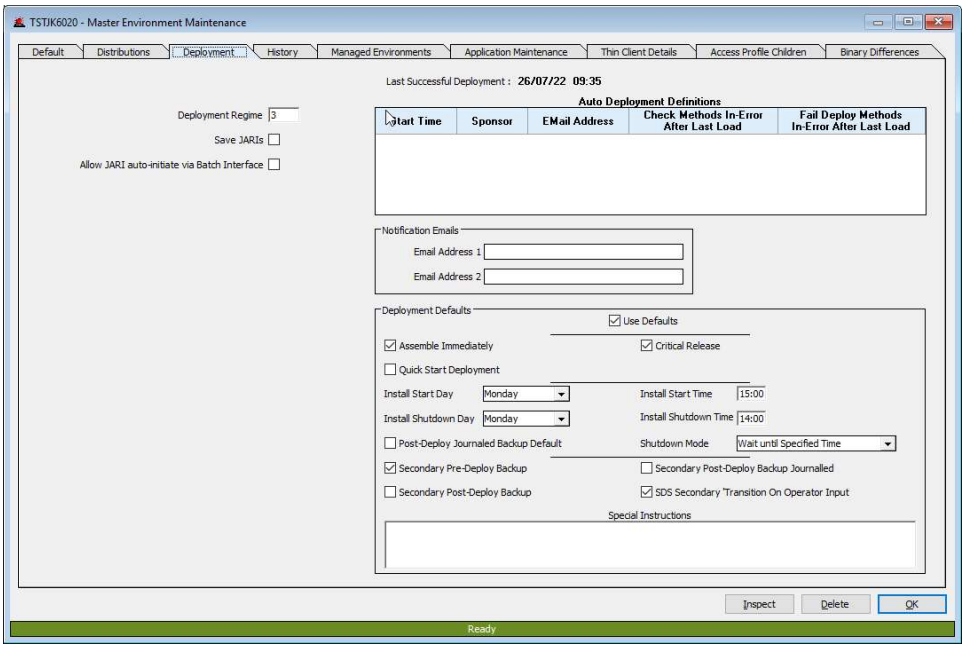


The Application Maintenance sheet not only defines the JADECare managed applications that have registered with JSM and provides pertinent details of each.

The sheet also provides right-click and menu options (below):

- to stop monitoring those applications that are marked as being monitored
- the deletion of inactive and non-monitored applications

Deployment

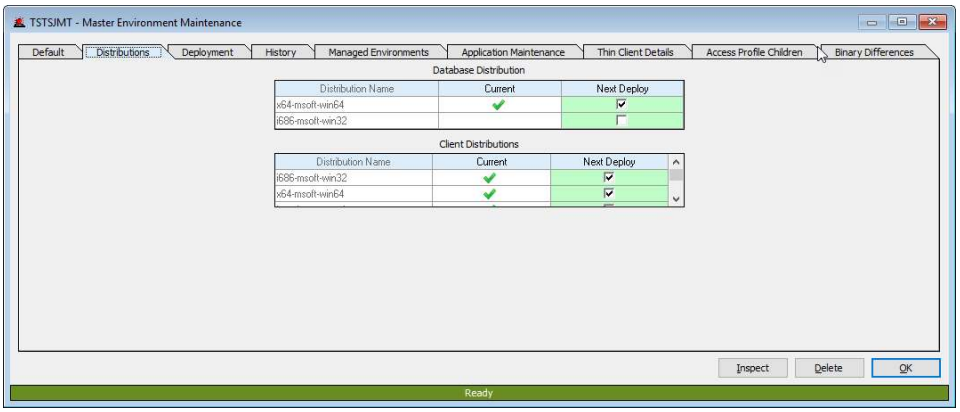


Auto Deployment Definitions	Multiple auto-deployment definitions can be maintained and are listed in the table. Right-clicking in the table will provide add, change and delete menu with the add and change displaying an Auto-Deployment Maintenance form, see below. Double-clicking on a row will cause the Change dialog to be displayed.

Deployment Regime	Defines which deployment regime is to be used. The default value is currently 2 and may not be less than 2.
Save JARIs for repeat deployment	Any JARI's used in the deployment process will be retained for future use and will be exempt from janitor process deletion.
Allow JARI auto-initiate via Batch Interface	Allow JARIs to be created via JossBatch application request startJariBatchBuild
Post-Deploy Journaled Backup Default	Specifies the default backup paradigm to be used for post-deployment backups (if the pre-requisites have been met)
SDS Secondary 'Transition on Opeator Input'	Defines the default setting for all SDS Secondaries 'Transition on Operator Input' when authorising JARI.
Notifications	Up o two email address notifications can be pre-defined. These will append and auto-fill the JARI notifications if not already defined.
Use Defaults	When this option is turned on, then deployments will use the defaults defined here. Any of these defaults can be overridden at runtime by the JARI authorisation forms.
Assemble Immediately	When true causes the Assemble Immediately to be defined. This value is ignored for JARI Multi-Approval.
Critical Release	When true causes the Critical Release checkbox to be defined
Quick Start Deployment	When true causes the Quick Start check box to be defined.
Install Start Day	Defines the preferred day of the week for when the installation sequence is to start and if pre-deploy back is not defeated, this will also be the backup start date.
Install Start Time	Defines the preferred time for the install to start.
Install Shutdown Day	Defines the preferred day of the week for when the system shutdown is to start.
Install Shutdown Time	Defines the preferred time of the day for the system shutdown to start.

Post-Deploy Journaled Backup Default	If the environment is capable of Journaled Backup, then when this is true, the Post-Deploy Journaled Backup will be set.
Secondary Pre-Deploy Backup	When true, sets the Secondary Pre-Deploy Backup.
Secondary Post-Deploy Backup	When true, sets the Secondary Post-Deploy Backup.
Secondary Post-Deploy Backup Journaled	If the environment is capable of Journaled Backup, then when this is true, the Secondary Post-Deploy Journaled Backup will be set.
SDS Secondary 'Transition on Operator Input'	When true, sets the Secondary 'Transition on Operator Input'
Special Instructions	When defined this will automatically be added to any new JARI and, when displayed, will be merged with any Developer Instructions.

Distributions

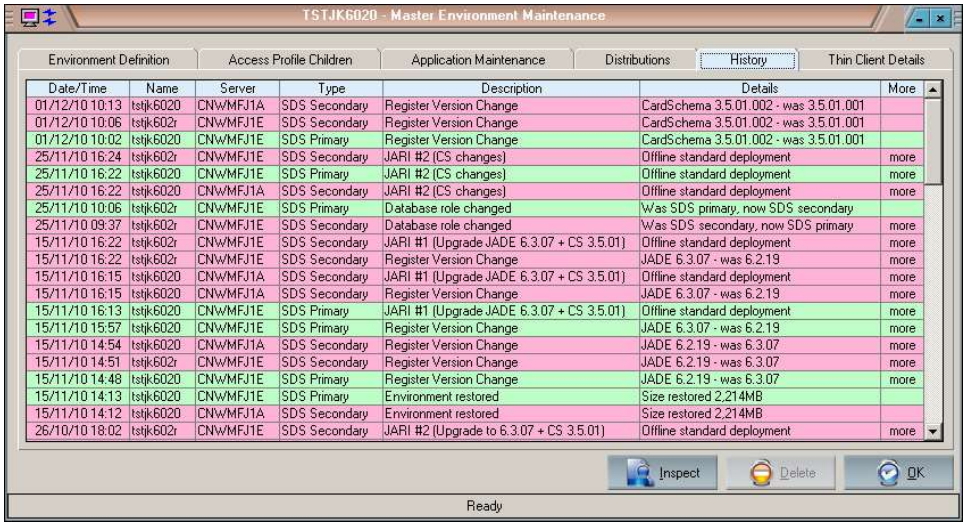


The 'Distributions' tab allows the user to associate distributions to be included in the next deployment. Only the distributions valid for the environment are shown.

The 'Current' column is read only and displays the current distributions the environment is using in deployment.

Distributions can be added or removed by manipulating the 'Next Deploy' column.

History



Date/Time	Name	Server	Type	Description	Details	More
01/12/10 10:13	tstjk6020	CNWMFJ1A	SDS Secondary	Register Version Change	CardSchema 3.5.01.002 - was 3.5.01.001	
01/12/10 10:06	tstjk602r	CNWMFJ1E	SDS Secondary	Register Version Change	CardSchema 3.5.01.002 - was 3.5.01.001	
01/12/10 10:02	tstjk6020	CNWMFJ1E	SDS Primary	Register Version Change	CardSchema 3.5.01.002 - was 3.5.01.001	
25/11/10 16:24	tstjk602r	CNWMFJ1E	SDS Secondary	JARI #2 (CS changes)	Offline standard deployment	more
25/11/10 16:22	tstjk6020	CNWMFJ1E	SDS Primary	JARI #2 (CS changes)	Offline standard deployment	more
25/11/10 16:22	tstjk6020	CNWMFJ1A	SDS Secondary	JARI #2 (CS changes)	Offline standard deployment	more
25/11/10 10:06	tstjk602r	CNWMFJ1E	SDS Primary	Database role changed	Was SDS primary, now SDS secondary	
25/11/10 09:37	tstjk602r	CNWMFJ1E	SDS Secondary	Database role changed	Was SDS secondary, now SDS primary	more
15/11/10 16:22	tstjk602r	CNWMFJ1E	SDS Secondary	JARI #1 (Upgrade JADE 6.3.07 + CS 3.5.01)	Offline standard deployment	more
15/11/10 16:22	tstjk602r	CNWMFJ1E	SDS Secondary	Register Version Change	JADE 6.3.07 - was 6.2.19	more
15/11/10 16:15	tstjk6020	CNWMFJ1A	SDS Secondary	JARI #1 (Upgrade JADE 6.3.07 + CS 3.5.01)	Offline standard deployment	more
15/11/10 16:15	tstjk6020	CNWMFJ1A	SDS Secondary	Register Version Change	JADE 6.3.07 - was 6.2.19	more
15/11/10 16:13	tstjk6020	CNWMFJ1E	SDS Primary	JARI #1 (Upgrade JADE 6.3.07 + CS 3.5.01)	Offline standard deployment	more
15/11/10 15:57	tstjk6020	CNWMFJ1E	SDS Primary	Register Version Change	JADE 6.3.07 - was 6.2.19	more
15/11/10 14:54	tstjk6020	CNWMFJ1A	SDS Secondary	Register Version Change	JADE 6.2.19 - was 6.3.07	more
15/11/10 14:51	tstjk602r	CNWMFJ1E	SDS Secondary	Register Version Change	JADE 6.2.19 - was 6.3.07	more
15/11/10 14:48	tstjk6020	CNWMFJ1E	SDS Primary	Register Version Change	JADE 6.2.19 - was 6.3.07	more
15/11/10 14:13	tstjk6020	CNWMFJ1E	SDS Primary	Environment restored	Size restored 2,214MB	
15/11/10 14:12	tstjk6020	CNWMFJ1A	SDS Secondary	Environment restored	Size restored 2,214MB	
26/10/10 18:02	tstjk602r	CNWMFJ1E	SDS Secondary	JARI #2 (Upgrade to 6.3.07 + CS 3.5.01)	Offline standard deployment	more

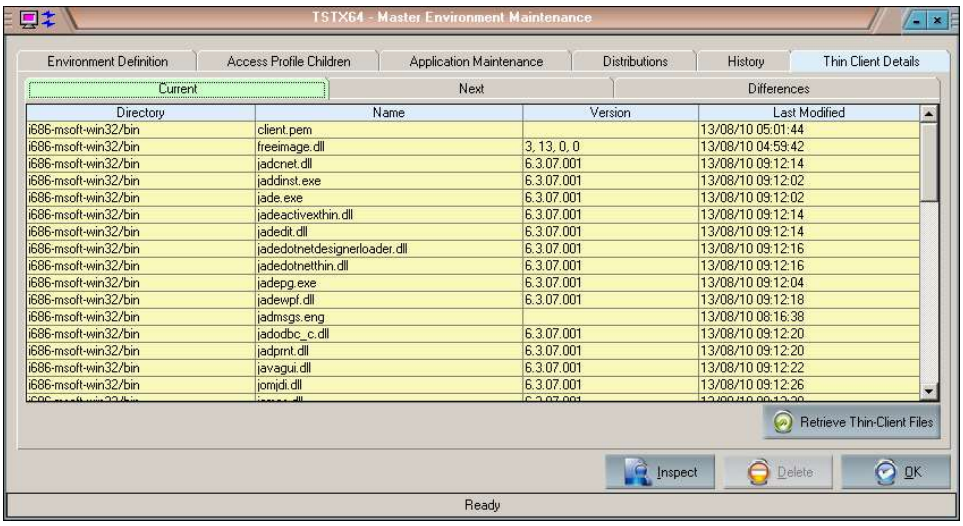
The History sheet provides a detailed history of significant events that have occurred for any of the host environments associated with the Master Environment.

Events such as :

- Deployments (and the content)
- JADE version changes
- CardSchema version changes
- Restores
- Cluster Failover
- SDS Takeover
- Host Environment enabled / disabled
- Host Environment deleted
- Host Environment created

This history is subject to archival as specified by the global retention period defined in the **System Global Options**

Thin Client Details



Directory	Name	Version	Last Modified
i686-msoft-win32/bin	client.pem		13/08/10 05:01:44
i686-msoft-win32/bin	freeimage.dll	3, 13, 0, 0	13/08/10 04:59:42
i686-msoft-win32/bin	jadcnet.dll	6.3.07.001	13/08/10 09:12:14
i686-msoft-win32/bin	jaddinst.exe	6.3.07.001	13/08/10 09:12:02
i686-msoft-win32/bin	jade.exe	6.3.07.001	13/08/10 09:12:02
i686-msoft-win32/bin	jadeactivexthin.dll	6.3.07.001	13/08/10 09:12:14
i686-msoft-win32/bin	jadedit.dll	6.3.07.001	13/08/10 09:12:14
i686-msoft-win32/bin	jadedotnetdesignerloader.dll	6.3.07.001	13/08/10 09:12:16
i686-msoft-win32/bin	jadedotnetthin.dll	6.3.07.001	13/08/10 09:12:16
i686-msoft-win32/bin	jadepg.exe	6.3.07.001	13/08/10 09:12:04
i686-msoft-win32/bin	jadewpf.dll	6.3.07.001	13/08/10 09:12:18
i686-msoft-win32/bin	jadmsgs.eng		13/08/10 08:16:38
i686-msoft-win32/bin	jadodbc_c.dll	6.3.07.001	13/08/10 09:12:20
i686-msoft-win32/bin	jadprnt.dll	6.3.07.001	13/08/10 09:12:20
i686-msoft-win32/bin	javagui.dll	6.3.07.001	13/08/10 09:12:22
i686-msoft-win32/bin	jomjdi.dll	6.3.07.001	13/08/10 09:12:26

The above sheet **Current** defines the thin-client files required for the master environment. This list is part of the regime 3 deployment and is maintained primarily by the deployment process.

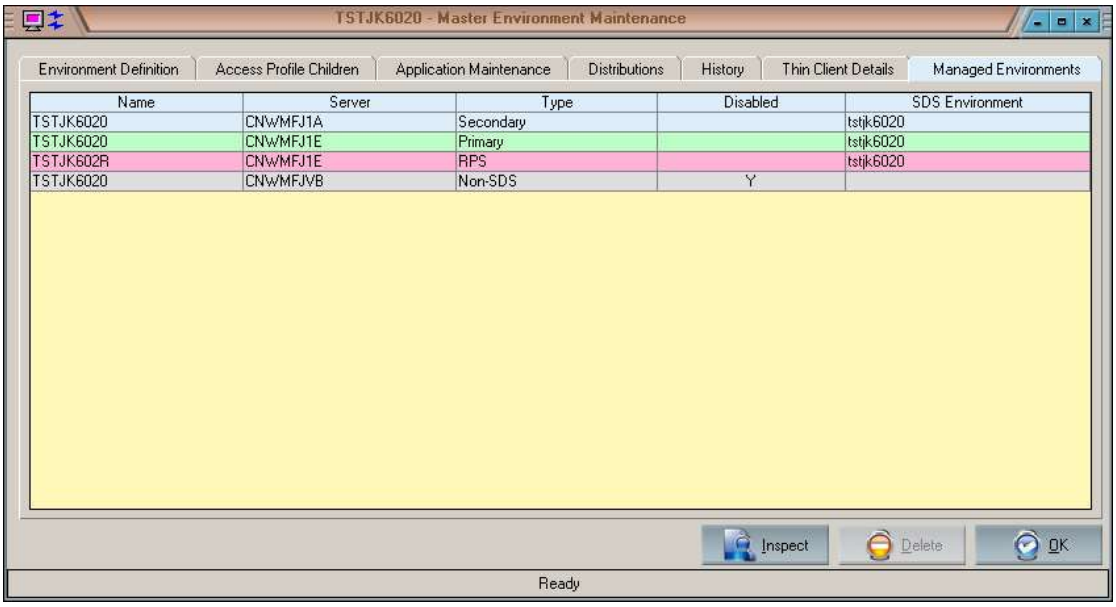
The **Next** and **Differences** sheet are used when a JARI has been created and the assembly created but has not yet been installed. These sheets are especially useful when the JARI regime 3 option **Thinclient Download Expected** has been set incorrectly causing the deployment assembly to fail.

The **Next** sheet shows all the thin-client files required as a result of the pending deployment.

The **Differences** sheet highlights what, if any, differences there are between the Current and Next (pending deployment) thin-client file sets.

The button **Retrieve Thin-Client Files** is required to be used before defining the deployment regime 3 or, only if required, to resynchronize JSM's master environment files with those actually deployed where these have changed for some reason (outside of JSM's control such as a manual restore).

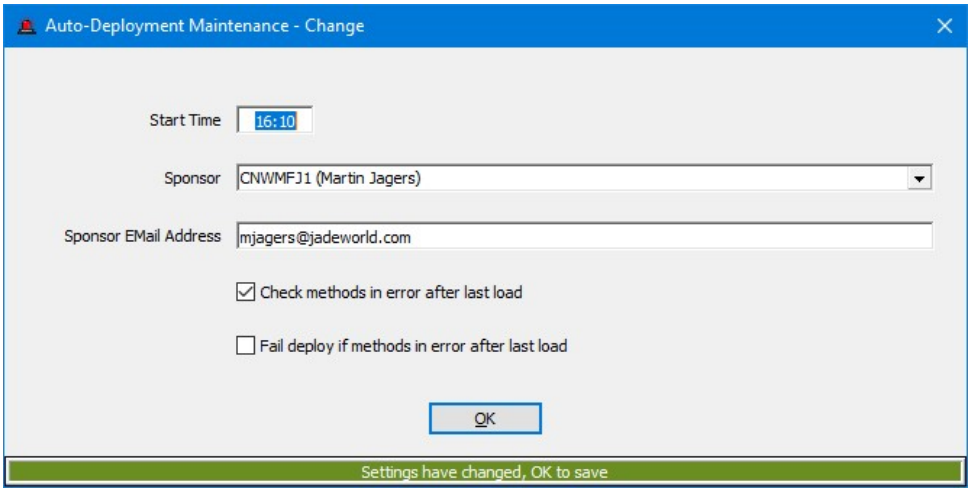
Managed Environments



The sheet shown above lists details of the managed environments (instances of the Master Environment) associated with this Master Environment.

Double-clicking on a row pertaining to one of the managed environments will cause the appropriate maintenance form to be opened (JADE Environment Maintenance or JOOB Environment Maintenance).

Auto Deployment Maintenance



Start Time	Defined in format hh:mm. NOTE : There must be two hours between each auto-deployment definition.
Sponsor	Must be a valid Operator entry
Sponsor Email Address	Can be used to override the default email address auto-filled on selection of sponsor above.
Check Methods in error after last load	Causes the deployment to check after the last schema entity has been loaded if there are any methods in error.
Fail deploy if methods in error after last load	If this option is checked then if there are any methods in error after the last schema entity has been loaded, then the deployment will fail otherwise the deployment will go to Hold state asking for operator input. This option is only valid if the Check Methods option above is set.

SNMP Tracker Interface

First implemented in release 3.8.06.

This interface requires the Tracker module to have been instantiated and JADECare utility cnwtrap.exe to be available.

The interface allows for SNMP traps to directly open and/or update Tracker calls and is designed to eliminate a large number of alerts requesting the Help Desk staff to open Tracker calls.

JADECare reserves specific trap numbers 900 to 999 for enterprise 1.3.6.1.4.1.2553, generic trap number 6 for use by JADECare Systems Manager's SNMP Interface.

At this juncture the following trap numbers are allocated :

- 900 – Tracker service for server host

- 901 – Tracker service for JADE database or remote
- 902 – Tracker service for ODBC database

Server Host – 900

The command line syntax is:

cnwtrap.exe 900 <server> <priority> <create|update> <alert > <short text > <long text> <assignee>

JADE Environment – 901

The command line syntax is:

cnwtrap.exe 901 <server> <env name> <priority> <create|update> <alert > <short text > <long text> <assignee>

ODBC Environment – 902

The command line syntax is:

cnwtrap.exe 902 <server> <env name> <instance name> <priority> <create|update> <alert > <short text > <long text> <assignee>

Parameters

Where arguments have embedded spaces they must be bounded by double quotes (“) for these to be successfully handled.

The following explains the arguments required to correctly implement the SNMP Tracker interface:

- <server> can be:
  - Fully qualified name - cnwxyz1a.cnw.co.nz
  - Simple server name – cnwxyz1a
  - IPv4 address – 131.21.21.131 – will use reverse lookup to resolve to DNS name
  - IPv6 address – 2029:8f71:85e2::7a61:71f9 – will use reverse lookup to resolve to DNS name
- <env name> - this is the name of the JADE Environment or ODBC Environment
- <instance name> - the name of the ODBC Instance which contains the ODBC Environment specified in <env name>
- <priority>
  - must be valid Tracker priority A, B, C or I
  - If the ‘update’ option is used and the assignee is specified then this may change the defined priority for the Tracker call which will affect escalation.
- <create|update>:
  - Use ‘create’ to always create a new Tracker call
  - Use ‘update’ to update the most recent Tracker call for the server where the Tracker call description matches the short text. Note that the Tracker call status must be ‘open’. If the status is ‘closed’ then a new Tracker call will be created.

- <alert>
    - Use ‘true’ to force an alert to be generated as well as open or update of a Tracker call
    - Use ‘false’ to suppress an alert and only open or update a Tracker call
- <short text>
    - free format short text, maximum allowed is 120 bytes
    - when ‘update’ option is used, this short text must match the Tracker call description (case-insensitive) so as to locate an existing open Tracker call
- <long text> - no limitation except what can be placed on the command line
- <assignee>
    - The assignee may define a valid user or security role or may be the literal ‘default’. If the ‘default’ literal is used this will:
        - use entity’s FMS Assignee if defined
        - If entity’s FMS Assignee is not defined will default to the ini file specified value for [JadeSystemsManager] OperationsRoleId.
    - If the ‘update’ option is used and the assignee is specified then this may change the defined assignee for the Tracker call.

Validation

When a SNMP Tracker trap is received, if any of the validation rules are violated, no Tracker will be created, rather a level 7 alert will be generated for each violation with short text ‘SNMP Tracker Interface Error’ and the long text will:

- Identify the nature of the error
- Provide details of each argument received

Example:

```
SNMP Tracker Interface 901 Error
CNWMFJ1D.cnw.co.nz  30 March 2017, 17:16:24

SNMP Tracker interface 901 expects minimum of 7 arguments

Number of trap variables = 6
- 1. server=no_such_server
- 2. env name=no_Jade_env
- 3. priority=C
- 4. create or update=Create
- 5. alert=false
- 6. shorttext=Too few arguments
```

Tracker plus Alert

If an alert is required as well as the creation or update of a Tracker call then the <alert> option should be set to true. In this case an alert will be generated for each trap that creates / updates a Tracker call where the alert will have short text ‘SNMP Tracker Interface Alert’ and will detail:

- Identify the entity:

- <server name>
- <JADE env name> on <server>
- <ODBC Env> in <ODBC Instance> on <ODBC Host>

- The Tracker call reference (created or updated)
- The 'short text' received in the trap
- The 'long text' received in the trap

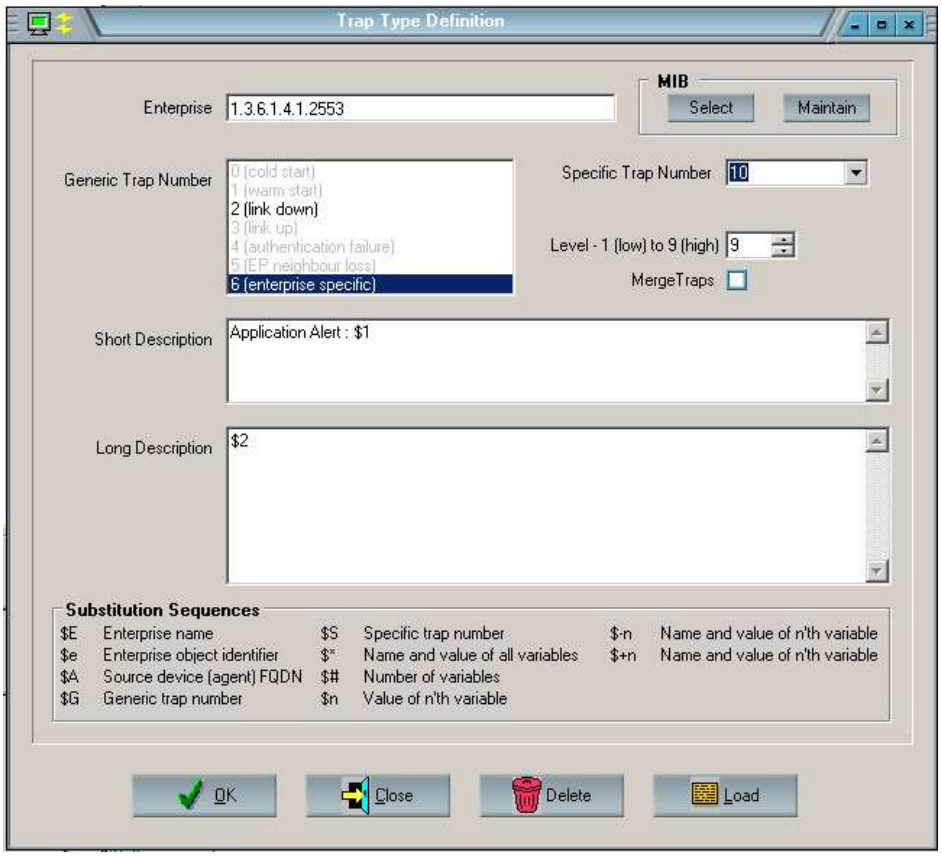
Example:

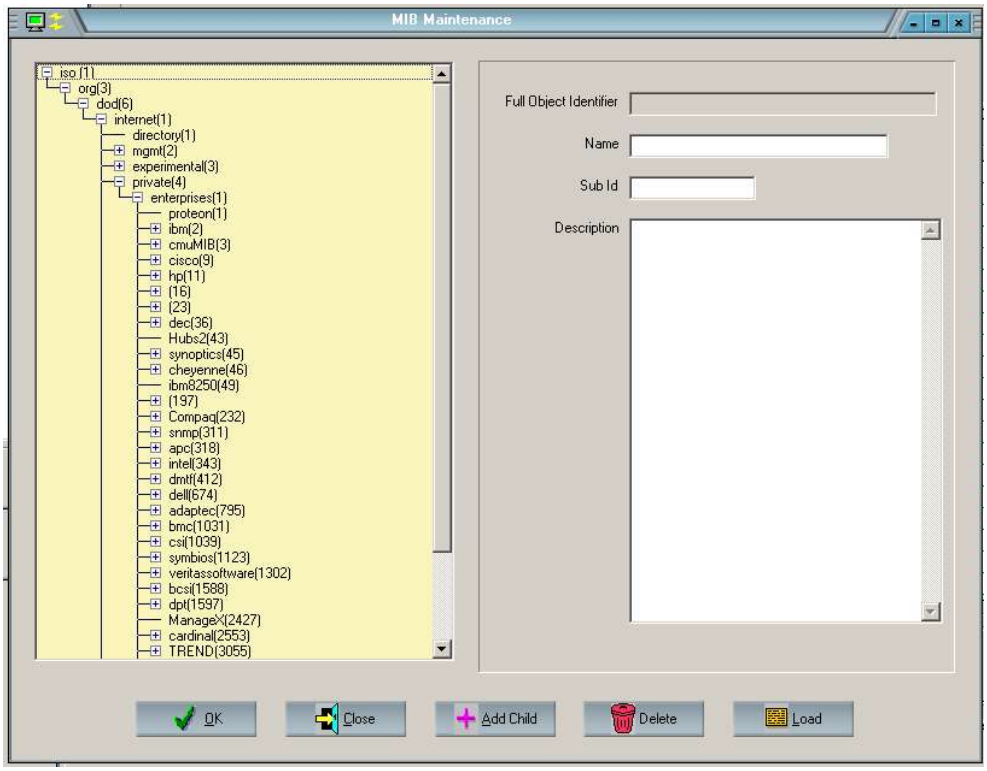
SNMP Tracker Interface 901 Error
CNWMFJ1D.cnw.co.nz 30 March 2017, 17:19:16
Error resolving IP Address '2002:8f60:83f9::8f60:83f9' to FQDN : The requested name is valid, but no data of the requested type was found.

Number of trap variables = 7
- 1. server=2002:8f60:83f9::8f60:83f9
- 2. env name=tst1unicd
- 3. priority=C
- 4. create or update=Update
- 5. alert=false
- 6. shorttext=Create new Tracker - short text
- 7. longtext=IPv6 address

Traps & MIBs

In order for JSM to decode SNMP traps (e.g. from managed servers, network devices or other hardware devices) and to raise JSM alerts, trap definitions need to be set up, and MIB files (Management Information Base) must be loaded into JSM using these forms.





When initializing an empty JSM database, a default MIB tree can be loaded from an extract taken from another system (e.g. JADE Software Corp's own JSM). The following two JadeScripts in the Joss schema can be used to achieve this, via the JADE development environment:

::mibTreeUnload	Extracts an existing MIB tree into a flat file
::mibTreeLoad	Clears any existing MIB data from the database and loads a MIB tree extract file.

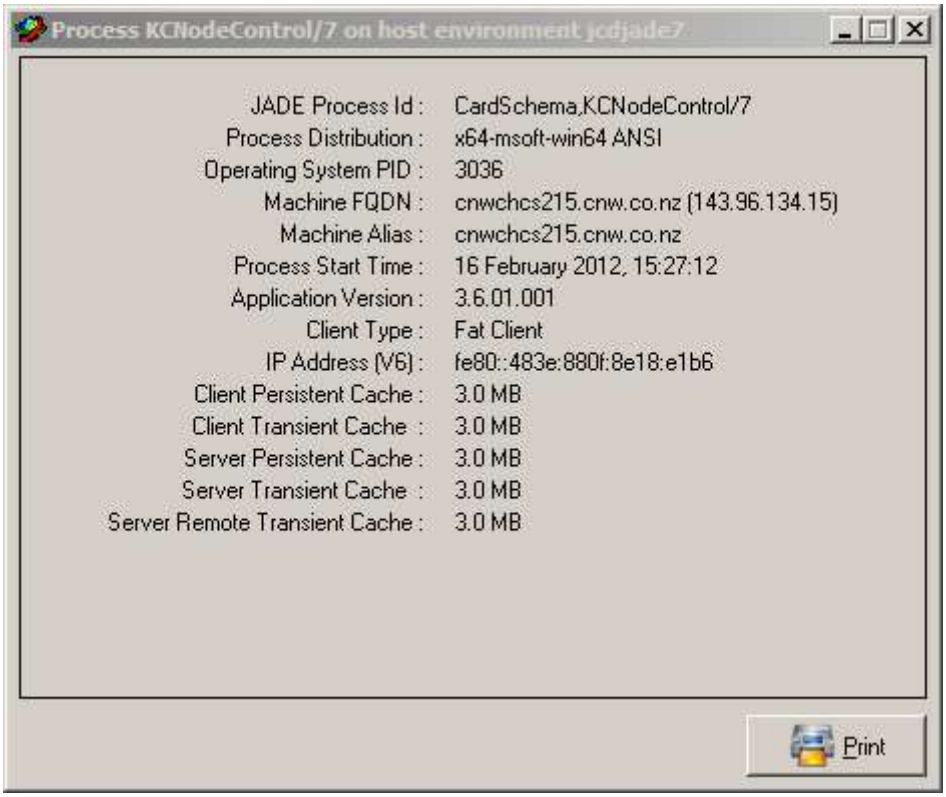
The above JSM form can then be used to load additional MIB files supplied by your hardware suppliers.

TCP/IP Address Handling

IPv6 support was introduced with the release of JADE 7. JSM and JSA support the increased IPv6 address size requirement and will display IPv6 information if available.

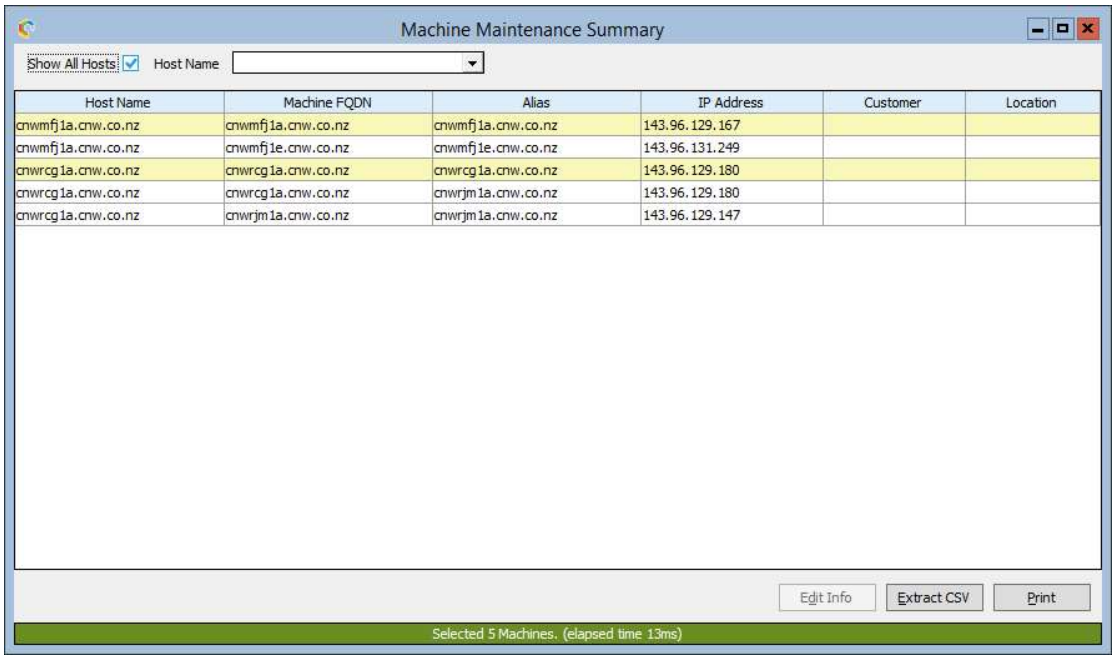
Note that the JADECare Systems Manager & JADECare Systems Agent products are not using IPv6 and will continue to use IPv4 however management of IPv6 enabled JADE environments is now possible.

For example in SystemView :


Process KCNodeControl/7 on host environment jcdjade7
JADE Process Id : CardSchema,KCNodeControl/7
Process Distribution : x64-msoft-win64 ANSI
Operating System PID : 3036
Machine FQDN : cnwchcs215.cnw.co.nz (143.96.134.15)
Machine Alias : cnwchcs215.cnw.co.nz
Process Start Time : 16 February 2012, 15:27:12
Application Version : 3.6.01.001
Client Type : Fat Client
IP Address (V6) : fe80::483e:880f:8e18:e1b6
Client Persistent Cache : 3.0 MB
Client Transient Cache : 3.0 MB
Server Persistent Cache : 3.0 MB
Server Transient Cache : 3.0 MB
Server Remote Transient Cache : 3.0 MB
Print

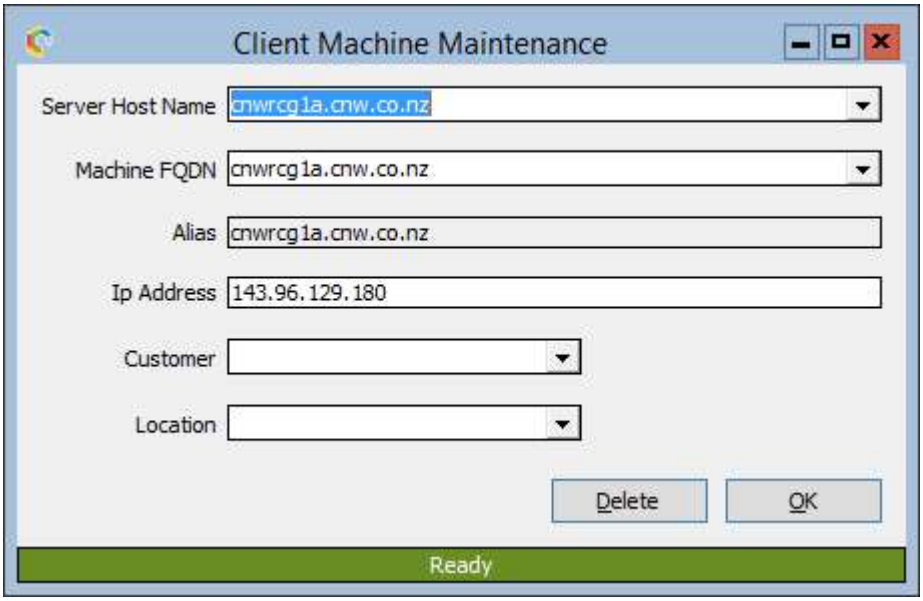
Machine Maintenance Summary

To view a summary of all machines click the ‘View’ menu item followed by ‘Machine Maintenance Summary’



Show All Hosts	Checking this will automatically show all machines.
Host Name	Selecting a host name will auto-uncheck the ‘Show All Hosts’ checkbox and only show machines relevent to the selected host name.

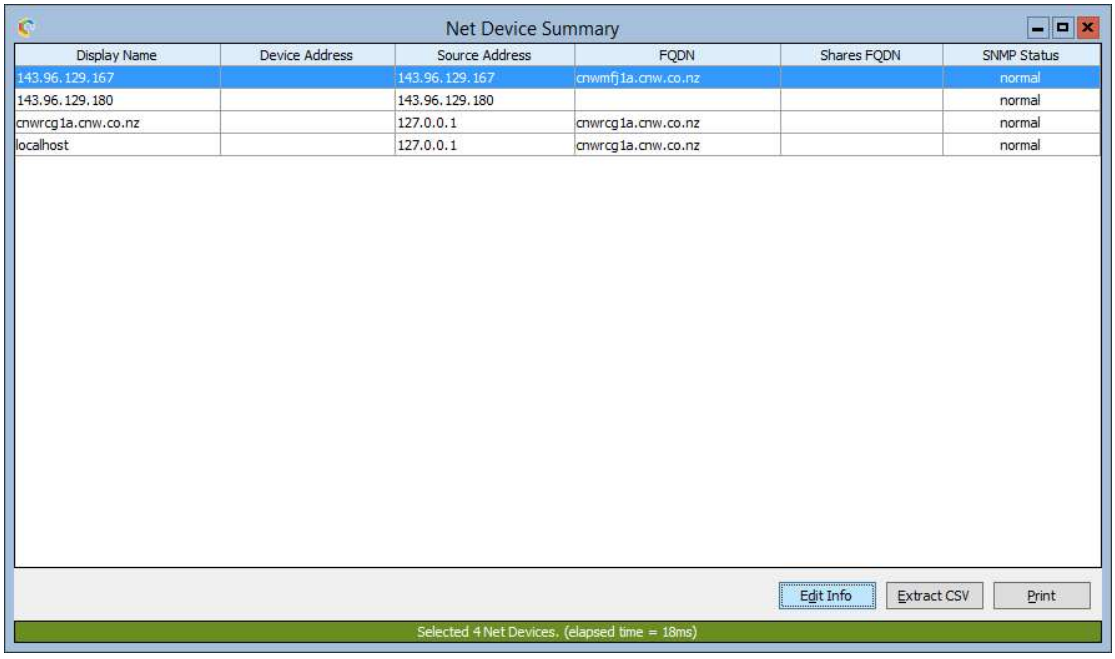
The bright yellow row denotes the current machine definition assigned to that host. Double clicking a row will bring up the machine maintenance form so the definition can be edited :



Changes made here will be reflected in the 'Machine Maintenance Summary' form automatically.

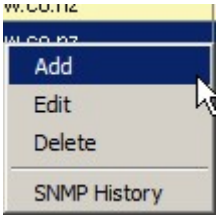
Net Device Summary

To view a summary of all net devices click the 'Setup/Traps and NIB' menu item then the sub 'Network Devices' menu item'.

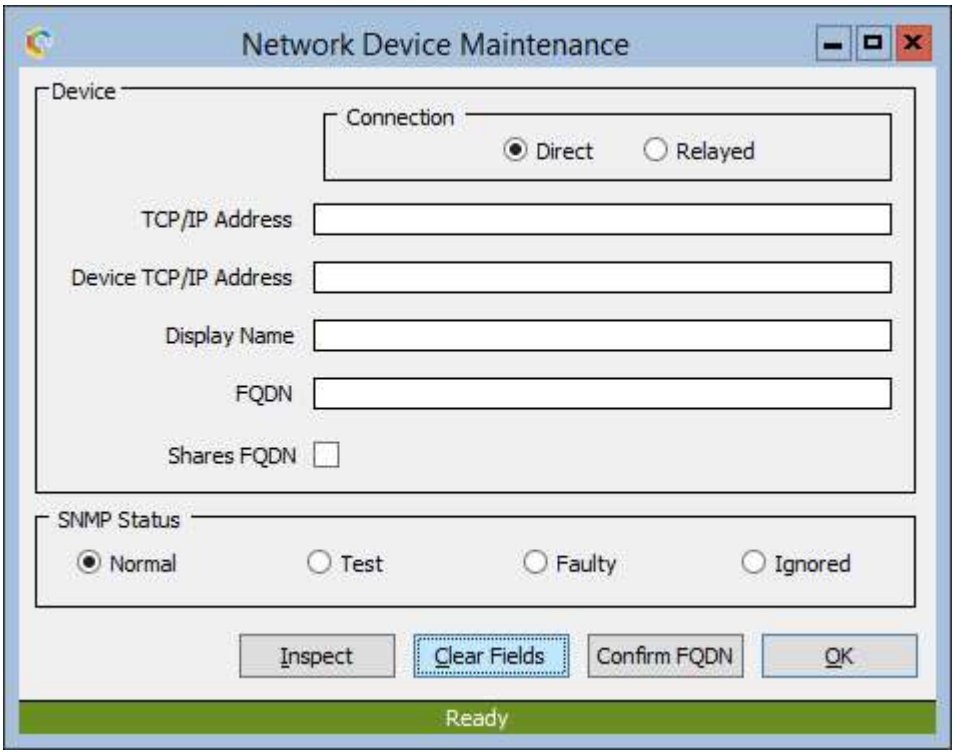


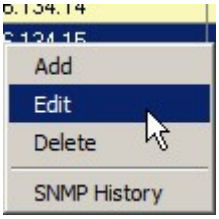
If the net device contains a 'Relay Address' then it is a relayed net device otherwise this is considered a direct net device.

This table contains a popup menu item which allows the addition, editing, deletion of a net device but also the display of any SNMP historical information.

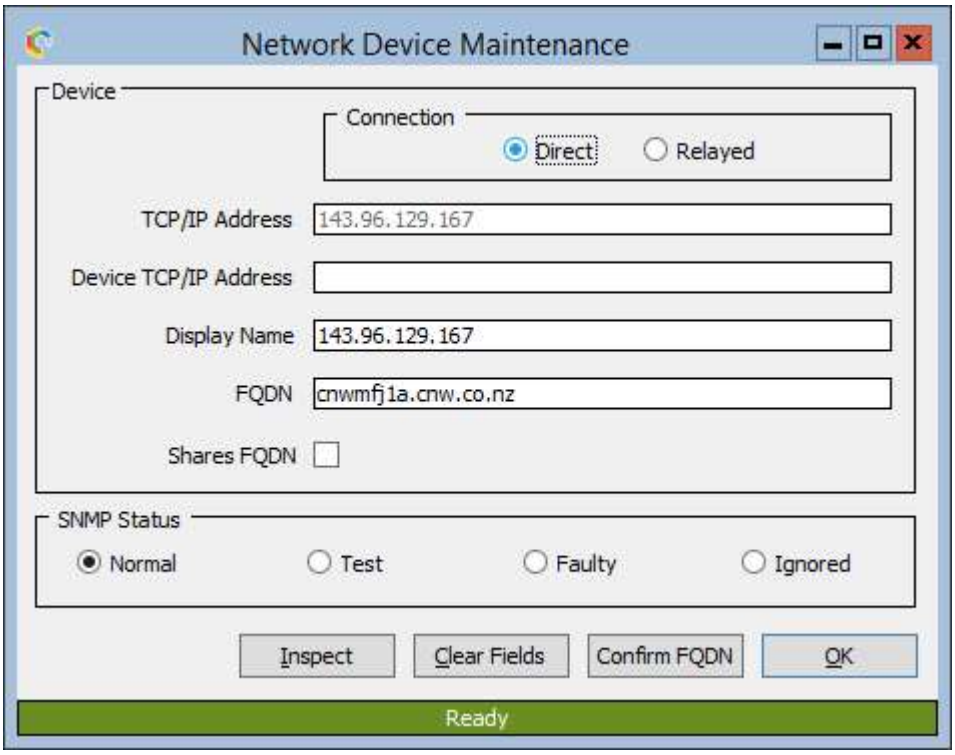


'Add' will display 'Network Device Maintenance' with all fields initialised :





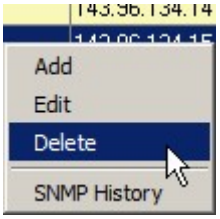
'Edit' or double clicking the selected entry will display 'Network Device Maintenance' with the selected entry information :



An already established network device cannot have the IP address information altered.

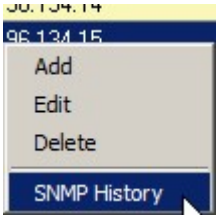
Display Name	A unique name that will be shown on forms to identify this device.
Device TCP/IP Address	The actual address of the device that generates SNMP traps.
Relay TCP/IP Address	The address of an intermediary device that forwards SNMP traps.
FQDN	Optional. Fully qualified domain name of either the device IP address or relayed IP address. If relayed IP address is defined then the FQDN should be the relayed IP address domain name.
Shares FQDN	Some FQDNs can have multiple IP addresses. If this definition fulls into that category then tick this box.
SNMP Status	Defines monitoring behaviour for SNMP traps generated by this network device.

Confirm FQDN	Performs a DNS reverse lookup on either the device IP address or relay IP address specified. If successful the FQDN is automatically filled in. Note that if both a device IP address and relay IP address are specified then the relay IP address is preferenced for the FQDN lookup.

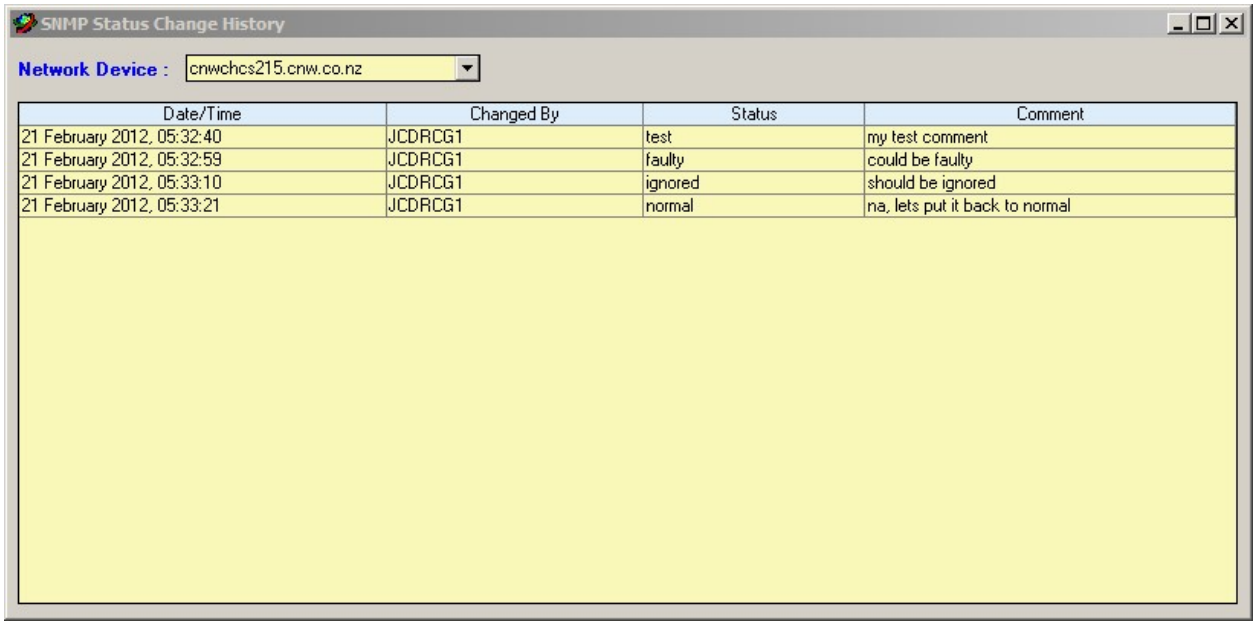


'Delete' on the selected entry will display a confirmation dialog :

'No' will cancel the request. Upon clicking 'Yes' some validation checks are performed to make sure the net device has no dependencies preventing it being deleted. If such dependencies exist the user will be notified and additional options will be presented.



'SNMP History' displays historical changes made to the SNMP status for the selected network device :



This information contains the change timestamp, operator who made the change, the status set during the change and any comment the operator made.

Server Definition

Server objects are normally automatically created (discovered) by JSM when the managed host's JSA connects to JSM. These objects may also be manually defined.

The fully qualified domain names of the server need to be appropriately defined in DNS so that JSM and JSA can correctly resolve the server name from the IP Address.

However some properties need to be set up manually, using the form shown below.

Fully Qualified Domain Name

Each server is known in JSM and JSA by a fully qualified name that is expected to be of the following format:

```
<computer name>.<domain name>
```

Where the first 3 characters of the computer name matches the Master Customer Id (as defined in the ini file) then the fully qualified name is made up as follows:

```
<computer name>.<Master Customer Domain Name from ini file>
```

Where the first 3 characters of the computer name does NOT match the Master Customer Id (as defined in the in I file) then the fully qualified name is made up as follows:

```
<computer name>.<first 3 chars of computername>.<Master Customer
Domain Name from ini file>
```

Examples:

Master Customer Id : XYZ

Master Customer Domain Name : xyz.com.aa

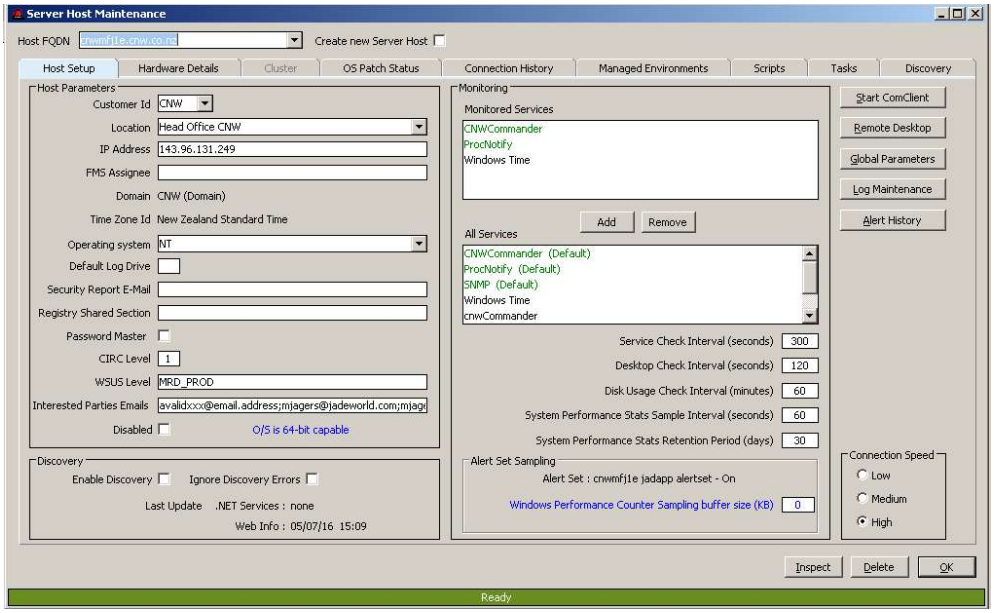
Computer Name: xyzmars1

The IP address would be expected to resolve this to **xyzmars1.xyz.com.aa**

Computer Name: abcmars1

The IP address would be expected to resolve this to **abcmars1.abc.xyz.com.aa**

Creating or Updating Host



Alert History Button

Clicking on this button will bring up the Alert History form prefilled with alert history details generated in the last 24 hours for the machines associated with this server.

Disabled Checkbox

Setting the disabled flag will cause the server host entry to be visible on the System View form only if the System View form has specified ‘Show Disabled Entities’.

If the server host has been marked as disabled, then if a JSA connection is established with that server host, then the disabled flag is reset.

Management Connection Speed:

This item is currently used by the deployment distribution facility and is used to determine the size of the blocks written to JSA.:

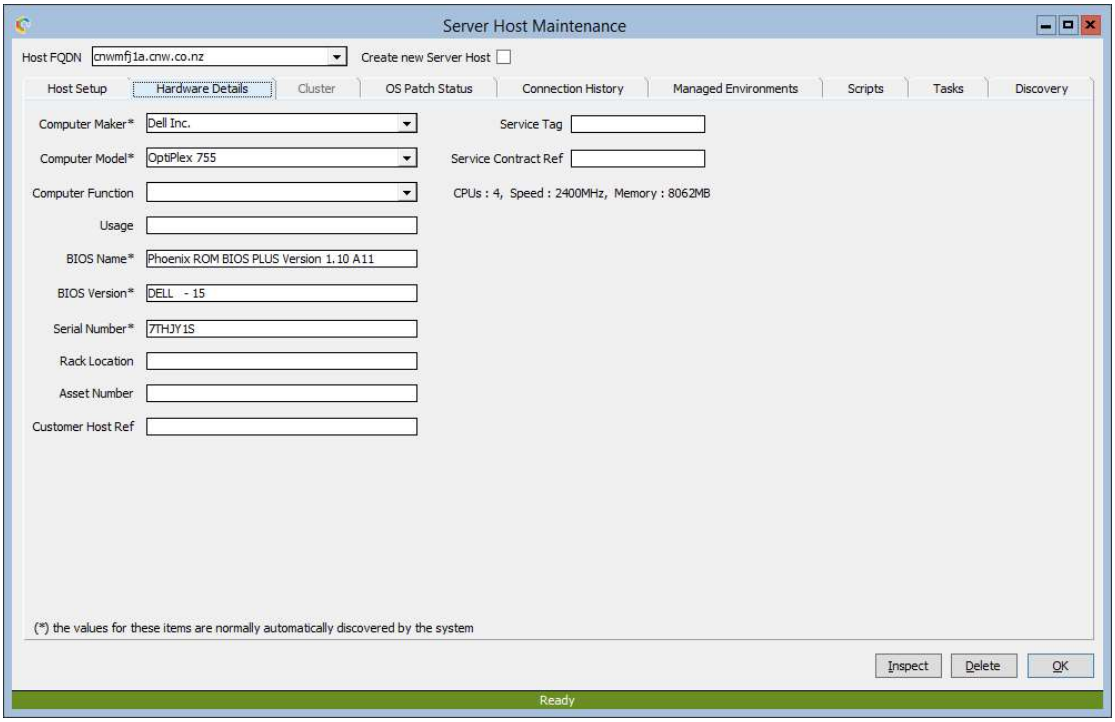
Slow	16,000 bytes
Medium	32,000 bytes
High	47,500 bytes

The default setting for a new host is High. The appropriate value is determined by the implementor of the tool set. The type of connection being used, the number of concurrent distributions (also affected by **Max File Transfer Sessions** in **Deployment** sheet of **System Global Options**), the impact on the JSM host, the usage of the target host and the distance need to be considered when changing this value.

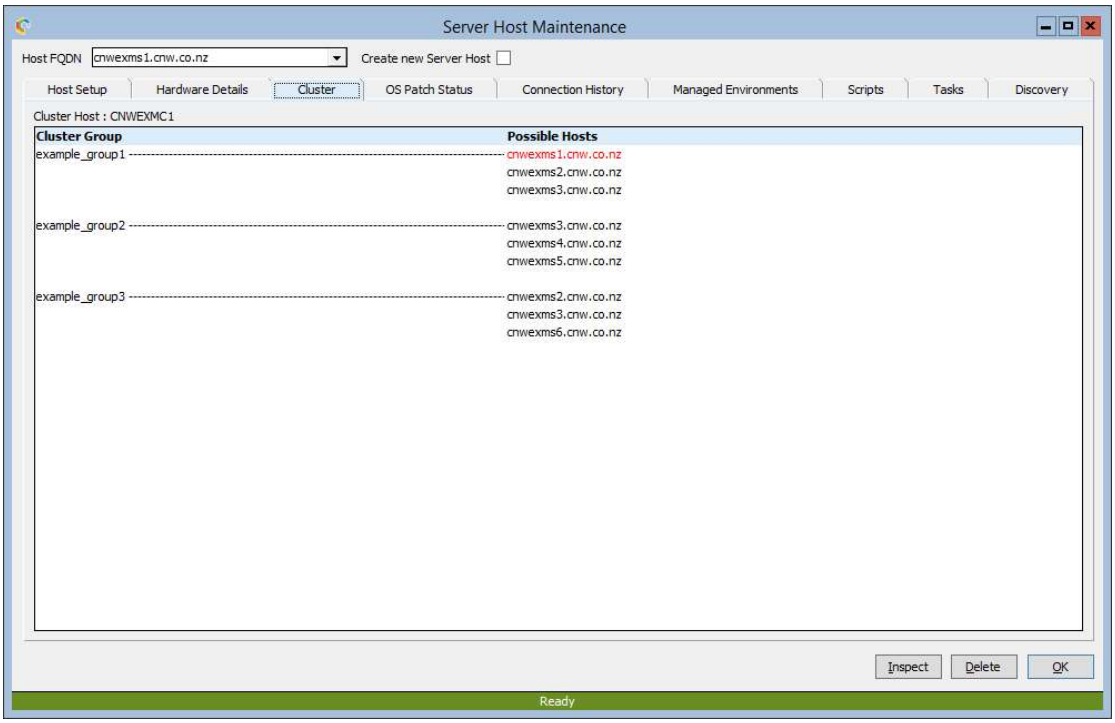
Disk Usage Check Interval:

The value entered here determines the interval at which JSA will sample the percentage of used disk space for each physical drive on the specified ServerHost. Collection of these statistics is used for the ‘Host Disk Usage’ report located under the System Reporting criteria.

Hardware Details



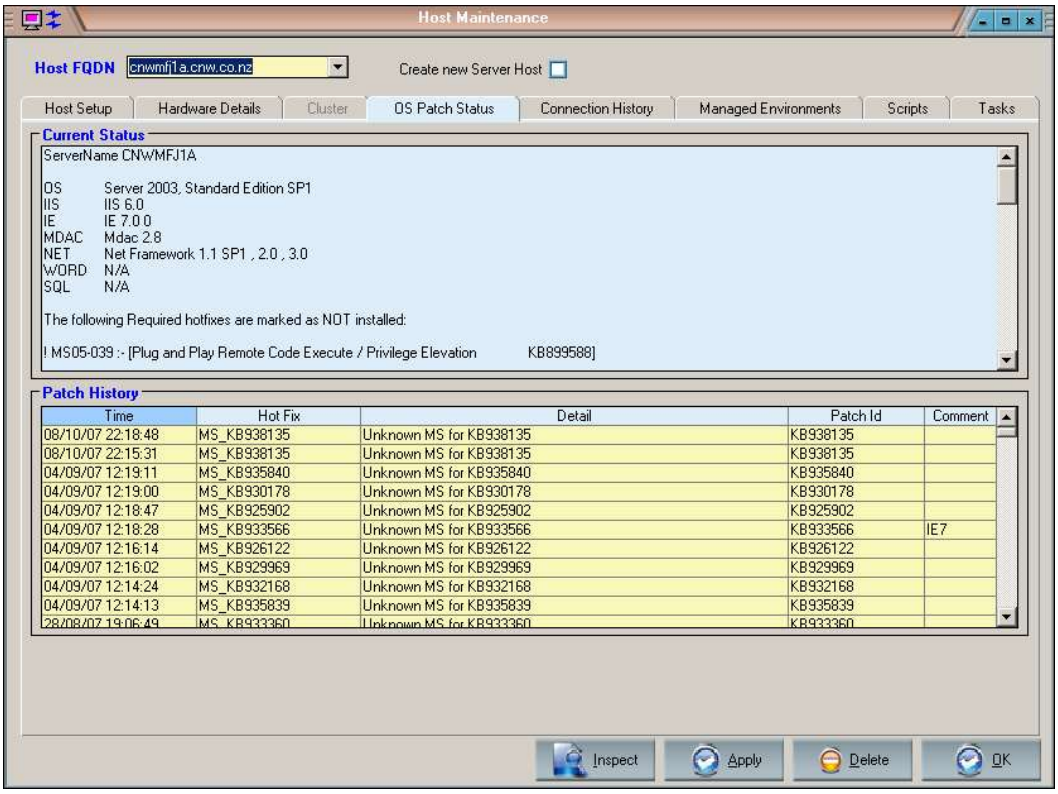
Cluster



If a server host is part of a cluster the sheet **Cluster** will be enabled and populated with cluser details such as the group names that this host belongs to and all the possible hosts that the group can use.

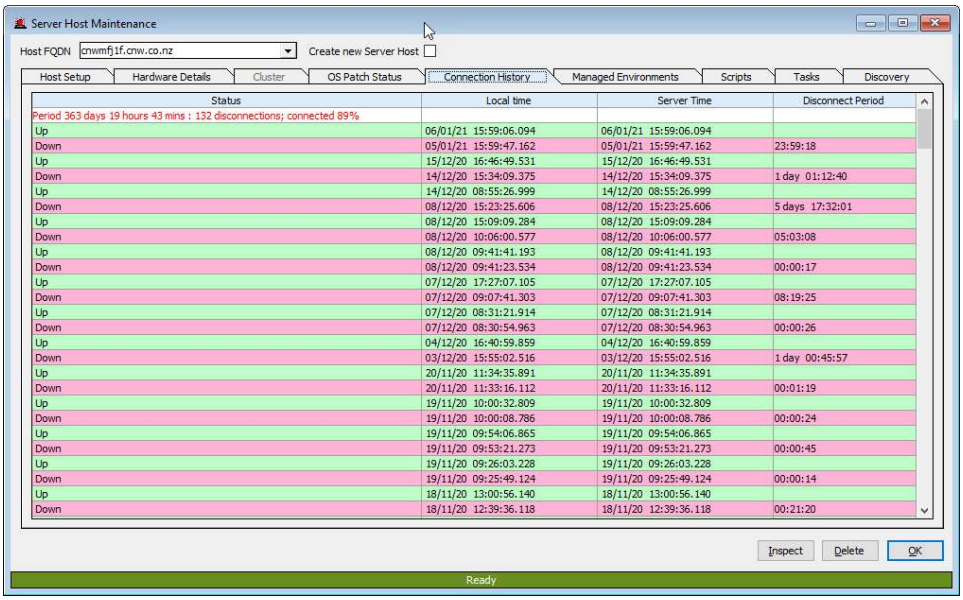
OS Patch Status

A history of Microsot patches is provided on the sheet **OS Patch Status**.



Connection History

The history of the connectivity from JSM to JSA running on the host is shown below.

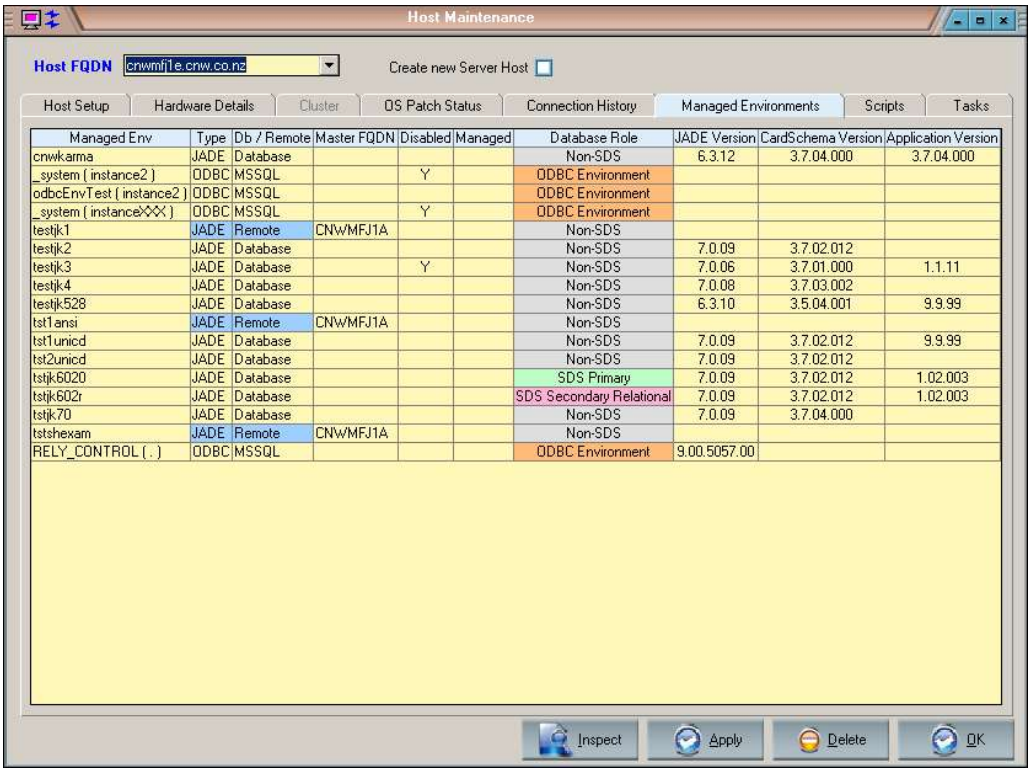


The first row will display:

- The period covered by the history displayed
- The number of disconnections for the period
- The percentage of time connected

Managed Environments

A complete list of the managed environments defined for this server host are listed with some status information. This includes any managed environments that are disabled.



Managed Env	Type	Db / Remote	Master FQDN	Disabled	Managed	Database Role	JADE Version	CardSchema Version	Application Version
cnwkarma	JADE	Database				Non-SDS	6.3.12	3.7.04.000	3.7.04.000
_system (instance2)	ODBC	MSSQL		Y		ODBC Environment			
odbcEnvTest (instance2)	ODBC	MSSQL				ODBC Environment			
_system (instanceXXX)	ODBC	MSSQL		Y		ODBC Environment			
testjk1	JADE	Remote	CNWMFJ1A			Non-SDS			
testjk2	JADE	Database				Non-SDS	7.0.09	3.7.02.012	
testjk3	JADE	Database		Y		Non-SDS	7.0.06	3.7.01.000	1.1.11
testjk4	JADE	Database				Non-SDS	7.0.08	3.7.03.002	
testjk528	JADE	Database				Non-SDS	6.3.10	3.5.04.001	9.9.99
tst1ansi	JADE	Remote	CNWMFJ1A			Non-SDS			
tst1unicd	JADE	Database				Non-SDS	7.0.09	3.7.02.012	9.9.99
tst2unicd	JADE	Database				Non-SDS	7.0.09	3.7.02.012	
tstjk6020	JADE	Database				SDS Primary	7.0.09	3.7.02.012	1.02.003
tstjk602r	JADE	Database				SDS Secondary Relational	7.0.09	3.7.02.012	1.02.003
tstjk70	JADE	Database				Non-SDS	7.0.09	3.7.04.000	
tstshexam	JADE	Remote	CNWMFJ1A			Non-SDS			
RELY_CONTROL (.)	ODBC	MSSQL				ODBC Environment	9.00.5057.00		

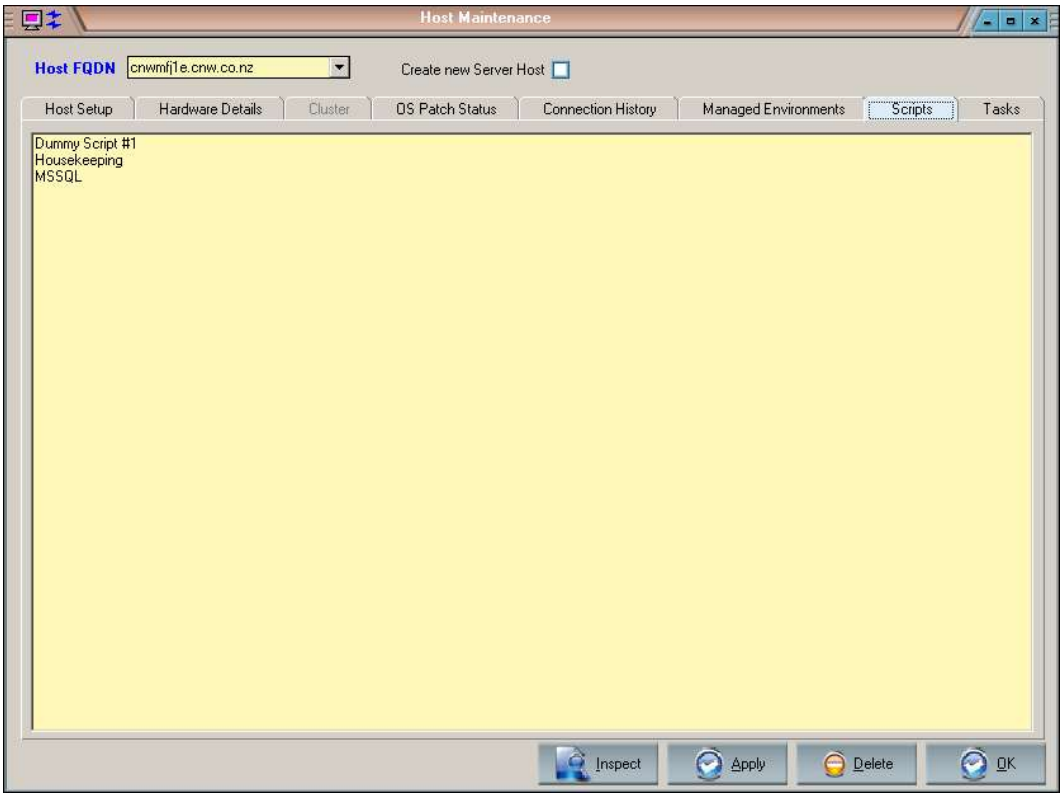
Double-clicking on a managed environment row will cause the appropriate maintenance form for the entity to be opened.

ODBC environments that are monitored by this server host but not managed by it will be appended to the bottom of the table. If the ODBC environment is managed then the server host FQDN for the managed server will be displayed.

Scripts

This sheet provides a list of all Task scripts being used by this server.

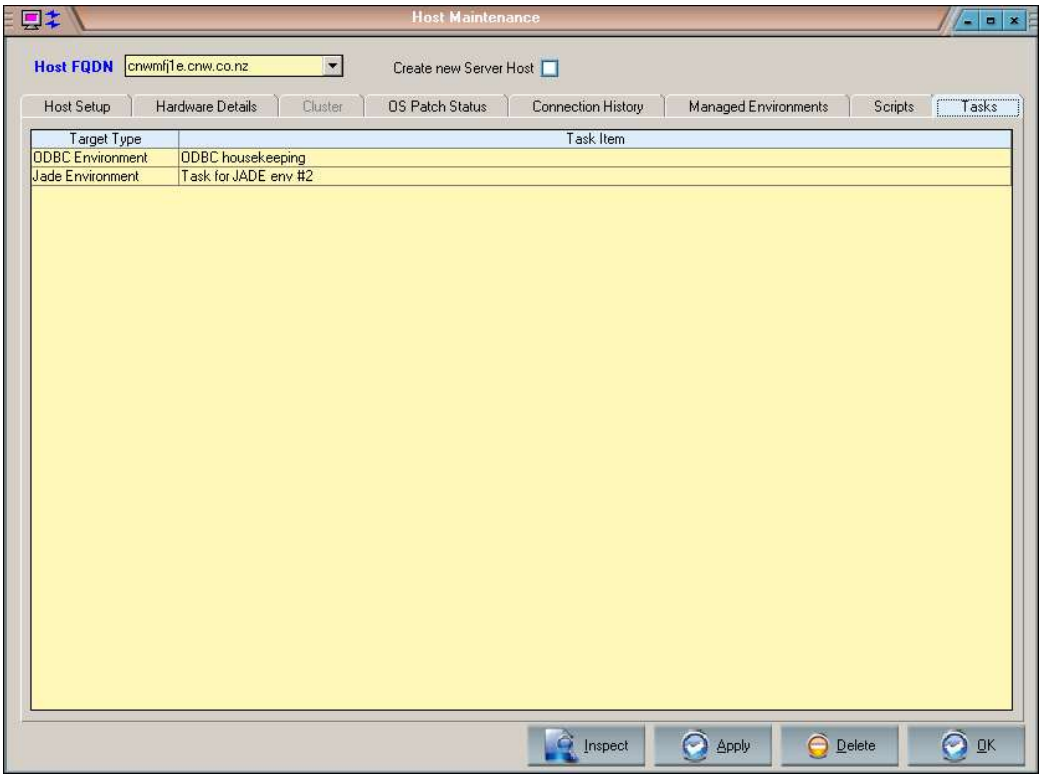
Double-clicking on a script entity will bring up the Script Maintenance form for this script.



Tasks

This sheet provides a list of all tasks associated with this server.

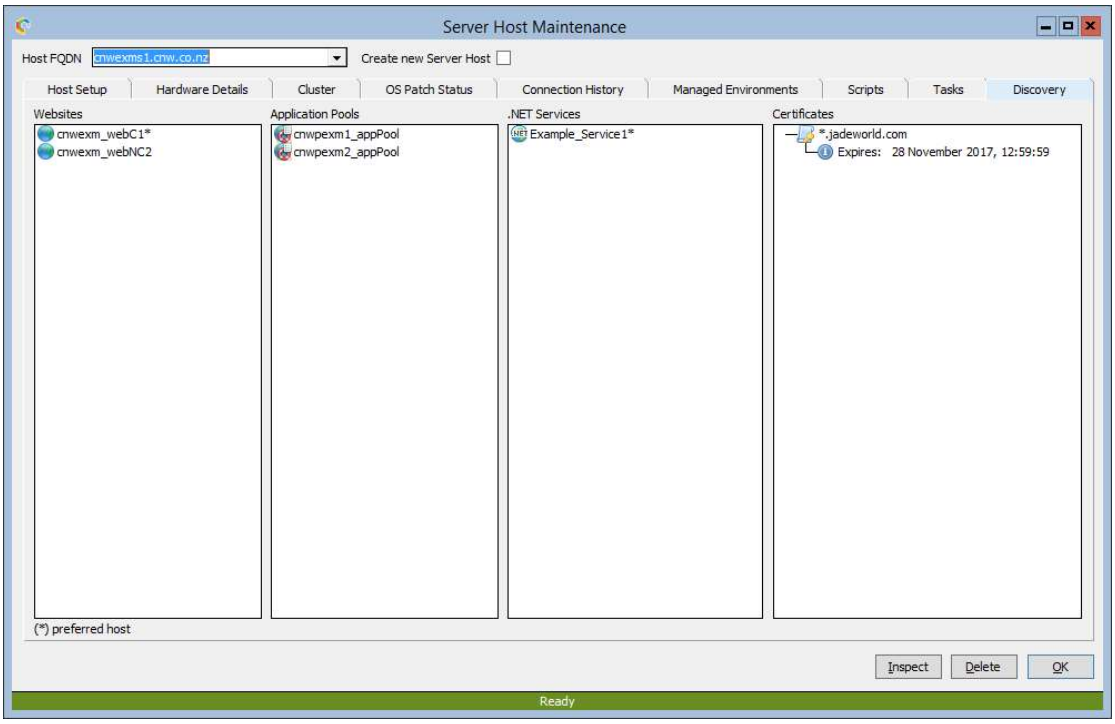
Double-clicking on a task will bring up the Task Maintenance form for this task.



Discovery

This sheet provides a list of all discovered Websites, Application Pools, .NET Services and Certificates associated with this server.

Double-clicking on an entry will bring up the appropriate form for the selection.



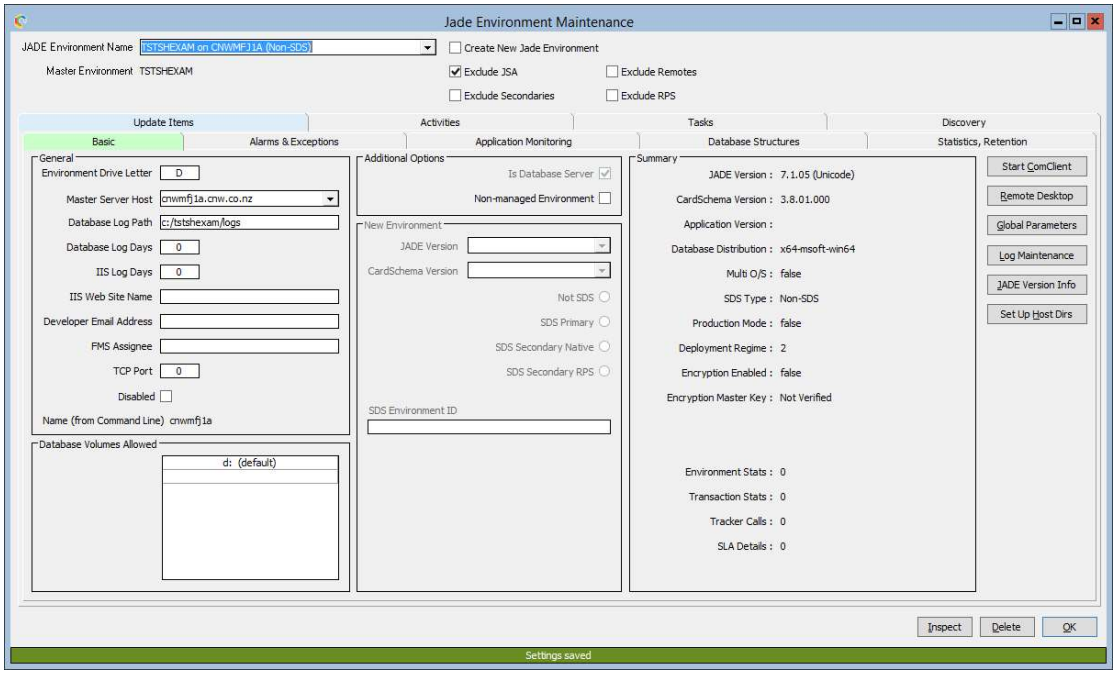
JADE Environment Maintenance

This form is accessible to most users but will be disabled for non-operational staff. It has 4 major sheets, some or all of which will be visible. Each of these major sheets, see below, also have multiple sheets, each being for a specific area / function

The major and minor sheets are listed below with an in-depth definition of each being provided later in this section:

- Update Items
  - Basic
  - Alarms & Exceptions
  - Application Monitoring
  - Database Structures
  - Statistics, Retention, Restrictions
- Activities
  - Backups
  - Compact
  - Restore
  - Relocate
- Cluster & SDS
- Thinclient

Basic sheet



Create New Jade Environment	Check this item to manually create a new Jade environment entity.
Exclude JSA	If this checkbox is ticked, then the JADE Environment Name combobox will NOT show any JADE Systems Agents (JSA).

	Uncheck this box to be able to select and show JSA managed environment details.
Environment Drive Letter	identifies the disk drive where the system's directories, binaries, database (if this is the database server) and files are to reside.
Master Server Host	the host where the database server (database host environment) resides.
Database Log Path	the root path for the database logs. This location is also used for the jommsg logs and the CnLogs that are provided by CardSchema.
Database Log Days	identifies how long the log files are to be retained for. The backup housekeeping process will remove any logs older than the number of days specified. If log days is zero, no housekeeping of log files will take place.
IIS Log Days	Not currently used by JSM, exported into Global Parameter files.
IIS Web Site Name	Not currently used by JSM, exported into Global Parameter files.
E-Mail Address	Not currently used by JSM, exported into Global Parameter files.
FMS Assignee	Not currently used by JSM, exported into Global Parameter files.
TCP Port	Not currently used by JSM, exported into Global Parameter files.
Disabled (check box)	Disables the host environment. This action leaves all statistical data and other parameters for the host environment intact, but removes it from display as a valid environment for the sleected host.. If the host environment is restarted then this flag is automatically reset.

Database Volumes Allowed

This table contains a list of the valid volumes on which database structures may exist. If any database structures exist on any volume other than those defined then an alert will be generated for each such structure whenever the database details are reported to JSM. The reporting of database details to JSM occurs:

- Approx 15 secs after KCCMgr start up
- Each backup
- Each online compaction
- Daily – at the time that KCCMgr was started.

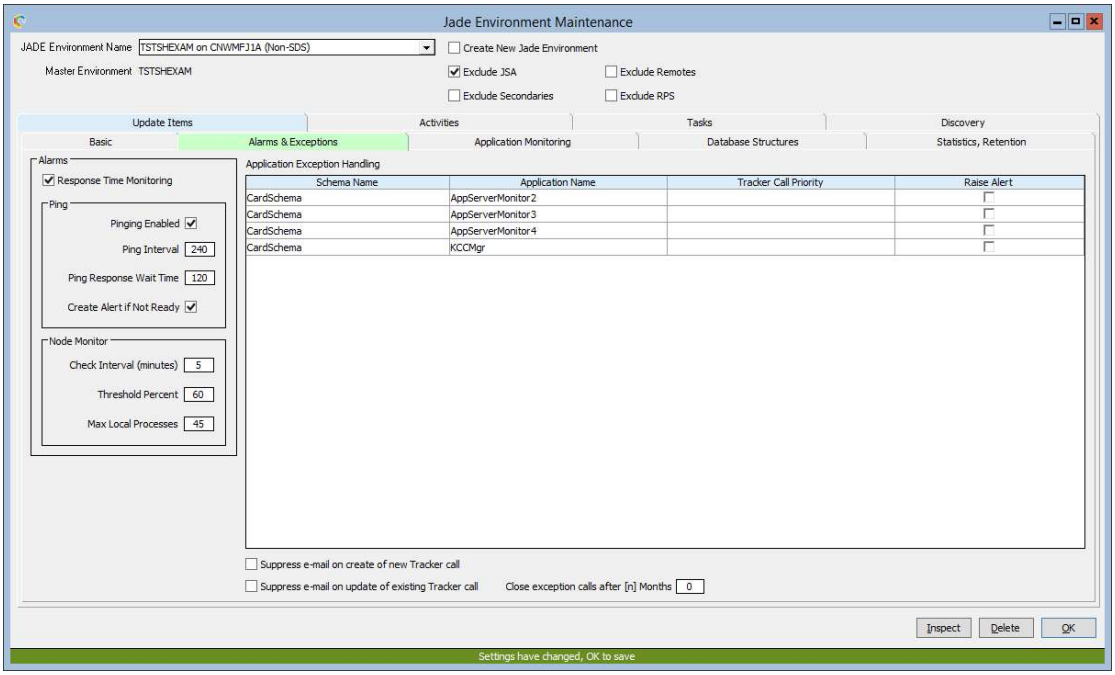
This sheet also provides the ability to create a new host environment entity manually, database or remote using the check box 'Create New Host Environment Object' which then causes the controls in the New Environment group box to be enabled.

The Summary pane lists summary data about the existing host environment.

Buttons for various key functions are provided as follows:

- Start Commander — starts up a cnwcomclient connection to the server where this host environment is resident
- Remote Desktop — starts up a Remote Desktop session to the server where this host environment is resident.
- Global Parameters — pops-up the Global Parameter File Maintenance form for this host environment
- Log Maintenance — pops-up the Log Deletion form for this host environment.
- Set Up Host Dirs — initiates the process to create host environment directories, if not available, and to propagate the version and initial dummy rar files to the respective directories.
- JADE Version Info — pops-up a form listing all the binaries and libraries currently deployed for this host environment (as at the time KCCMgr was started).
- SLA Detail — pops-up the SLA Documentation form relevant for this host environment's service.

Alarms & Exceptions sheet



Alarms

Response Time Monitoring — Enables / disables response time monitoring module. This may only be set if the transaction statistics gathering interval (JOM / Tran stats Interval (secs)) is other than zero

Ping

Pinging Enabled (check box)	If reset then initialises all the pinging values for the host environment. If set updates the pinging values from these screen items.
Ping Interval	Defines the period (in seconds) to wait between checks that the host environment is active and answering. If set to zero, then the default value defined in **System Global Options** is used. This also changes the Ping Response Wait Time to the default value.
Ping Response Wait Time	The time elapsed in seconds that JSA is to wait for a response from the application before issuing an alert and/or audible alarm (depending on settings of Audible Alarm if Not Ready and Create Alert if Not Ready)
Create Alert if Not Ready	If set then should the host environment not respond to a ping within the specified Ping Response Wait Time then an alert is created in JSM

.

Node Monitor

The values below are inherited from the global options when a new host environment is created and is only activated for the host environment when the global option **Node Monitor Active** has been set and all values below are greater than zero.

Check Interval (minutes)	The time interval between checks (in minutes)
Threshold Percent	The percentage threshold of max local processes (as determined for the node) above which to alert.
Max Local Processes	This is the default value to use if the fat-client ini file setting doesn't have MaxLocalProcesses defined

Application Exception Handling

Applications using CardSchema 3.0 or later will now create or update JADECare Tracker faults if JADECare Tracker has been implemented and the System Global Options 'Report application exceptions to Tracker' check box is set.

Unhandled exceptions in applications no longer generate an alert. Rather this causes an exception advice to be sent to JSM.

A new call will be created if no open call already exists for this exception type for this application. When a new call is created, the priority will be B, this initial priority may be defined above.

An alert may be generated for action by the Central Help Desk, if an alert is specifically requested for this application. The default action is NOT to generate an alert.

Application Monitoring Sheet

In order for an application to be monitored, it must have been executed at least once and registered by activating Karma Control for it to be known to JSM, otherwise it will not appear in the table above.

Application monitoring is only defined on the database server host environment entity. The MinCopies setting applies to the total number of copies of a running application regardless of whether it is running local to the database server (on the same server host as the database node) or remotely.

Defining values on one or more of the known applications will cause the relevant JSA system to start monitoring, the specified applications and ensure they have the expected number of copies running and optionally, check that the applications are responding,.

If the specified number of copies are not running then an alert will be generated with level 9 with heading 'Critical process <schema>,<application> of <env name> on <server> has insufficient copies' and will provide details of how many copies are running.

Further, if the 'Ping Required' check box is set, each matching application instance that is running will be 'pinged' and a response expected within the 'Ping Wait Time' interval. If a response is not returned in the specified 'Ping Wait Time' then a level 7 alert with heading 'Critical process <schema>,<application> of <env name> on <server> not responding'.

Notes:

If the application monitoring is turned off, any 'Insufficient Copies' and 'Not Responding' alerts for the application concerned will be automatically cleared with update 'Application monitoring stopped'.

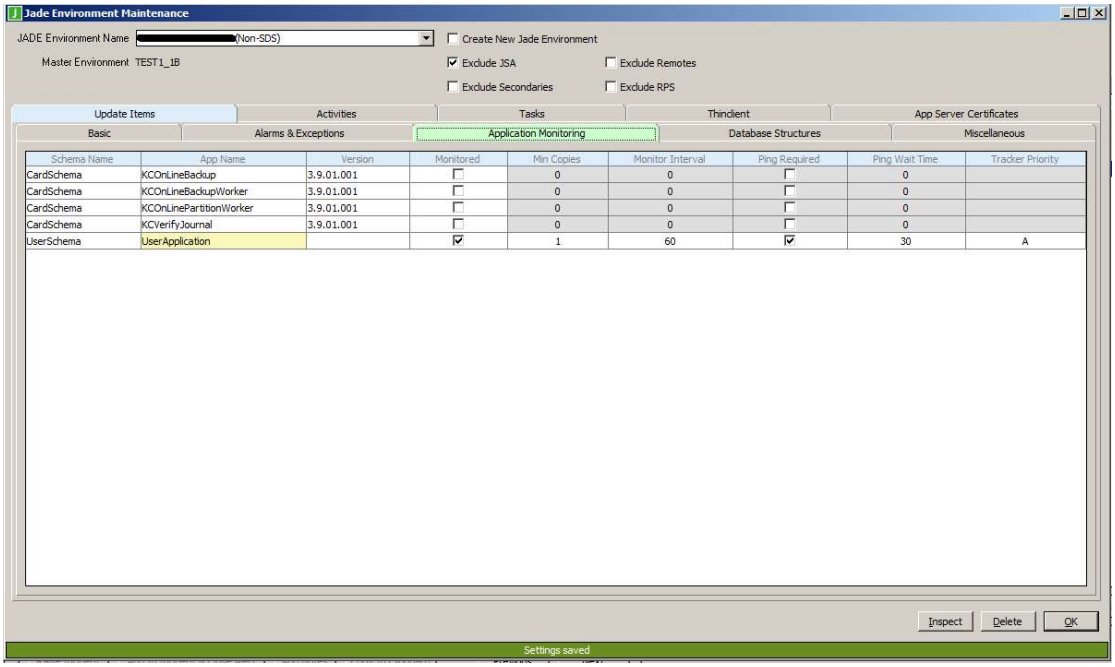
If the application monitoring is interval is changed, any 'Insufficient Copies' and 'Not Responding' alerts for the application concerned will be automatically cleared with update 'Application monitoring interval changed'.

If the JADE Environment for the applications being monitored terminates, any 'Insufficient Copies' and 'Not Responding' alerts for all of the applications of the JADE Environment will be automatically cleared with update 'Outstanding alert cleared due to database termination'.

If the JSA managing the JADE Environments terminates, any 'Insufficient Copies' and 'Not Responding' alerts for any of the JADE Environments will be automatically cleared with update 'Outstanding alert cleared due to JSA termination'.

If an application terminates that is being monitored and has 'Ping Required' set, any 'Not Responding' alerts will be automatically cleared with 'Process terminated'.

Non-SDS Jade Environment



The above applies whether the host environment is clustered or not.

For monitoring to be active, ensure a tick is visible in the Monitored check box.

Specify ‘Min Copies’, the ‘Monitor Interval’ in seconds , if ‘Ping Required’ is to be set and the ‘Ping Wait Time’.

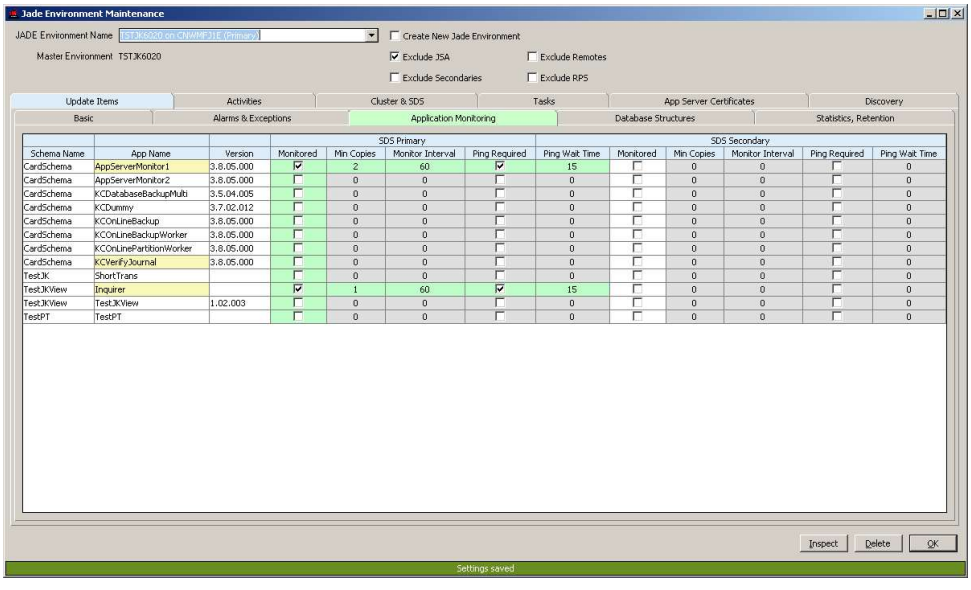
Note:

- the minimum Ping Response Wait Time is 5 seconds.
- The monitor interval must be more than 3 seconds greater than the Ping Wait Time.

Once your changes have been made, click on the OK button to effect this change and any other informational items shown on this form.

SDS Jade Environment

For an SDS environment the application monitoring sheet looks like this



The form above shows both the SDS Primary and SDS Secondary values for the **CURRENT** host environment on the one form. This allows you to define different monitoring settings depending on the SDS role this host environment is currently defined as. JSA will automatically switch to the appropriate monitoring settings as the role changes.

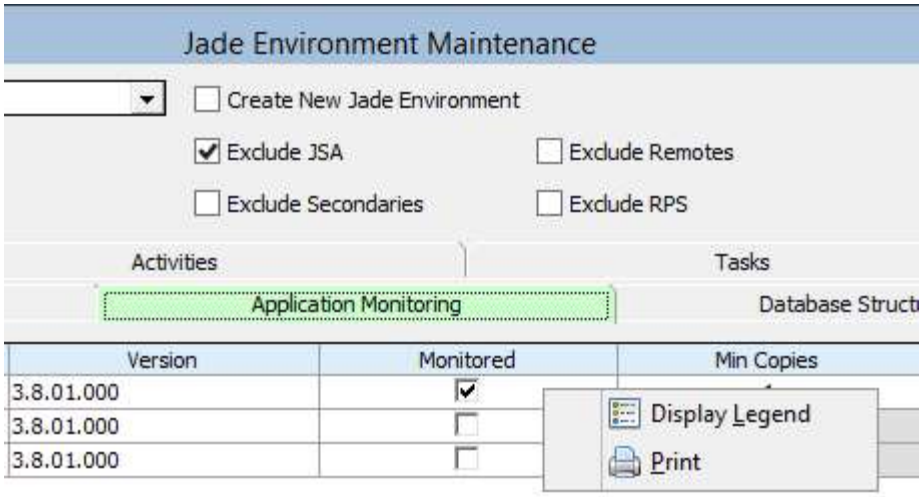
The green signifies the current role held by the target environment.

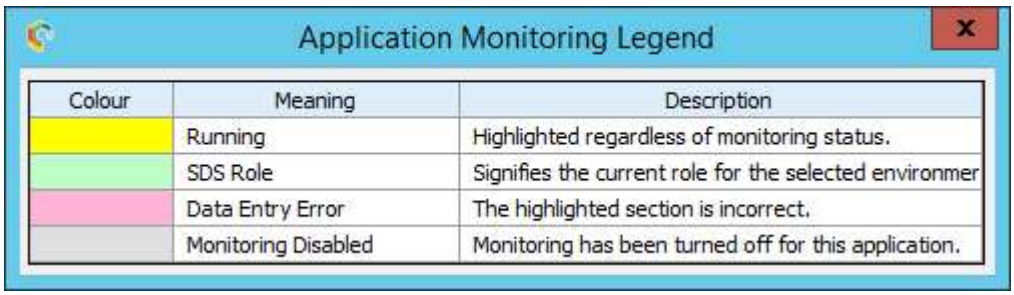
The gray signifies monitoring is currently disabled.

When an application is running the application 'App Name' entry is highlighted by bright yellow.

Should a data entry error occur the values that failed to validate are highlighted in pale pink.

These colour codes and their meanings can be viewed by a popup menu item on the table





Clustered Jade Environment

For a clustered environment the 'Application Monitoring' sheet looks the same as that of a non-SDS JADE environment or SDS JADE Environment (if SDS enabled).

The application monitoring values declared for a cluster are defined for the environment on the 'database server' only. This is a new feature introduced in JSM 3.5.01 as previously each cluster node may have had to have separate application monitoring defined.

In JSM 3.5.01 or later, one set of application monitoring definitions will 'follow' the cluster i.e.: if a cluster fail-over occurs the application monitoring will automatically switch to the new server where the database is running.

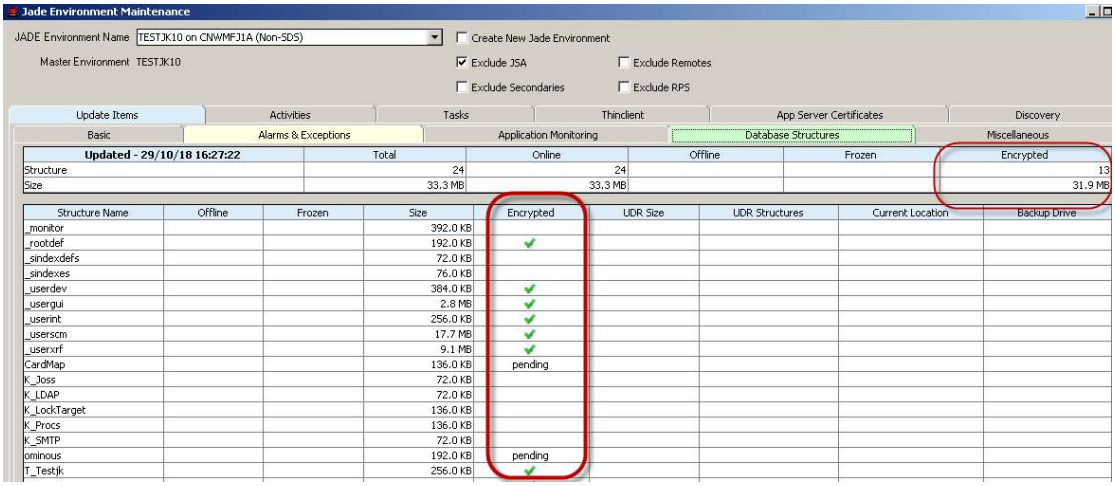
One of the important aspects of this change involves the ‘Min Copies’ value. This value no longer determines the number of copies running on a specific server but the number of copies running in total for the host environment.

If the number of applications running is less than that defined for ‘Min Copies’ and alert will be raised and, in the case of a clustered environment, the alert will contain information about where the “missing” application was last running. For example

Database Structures sheet

When the host environment is running on CardSchema 3.5.01 or later then the database structure details below will be shown.

Note that the UDR structures column is only visible when CardSchema 3.5.02 or later has been deployed.



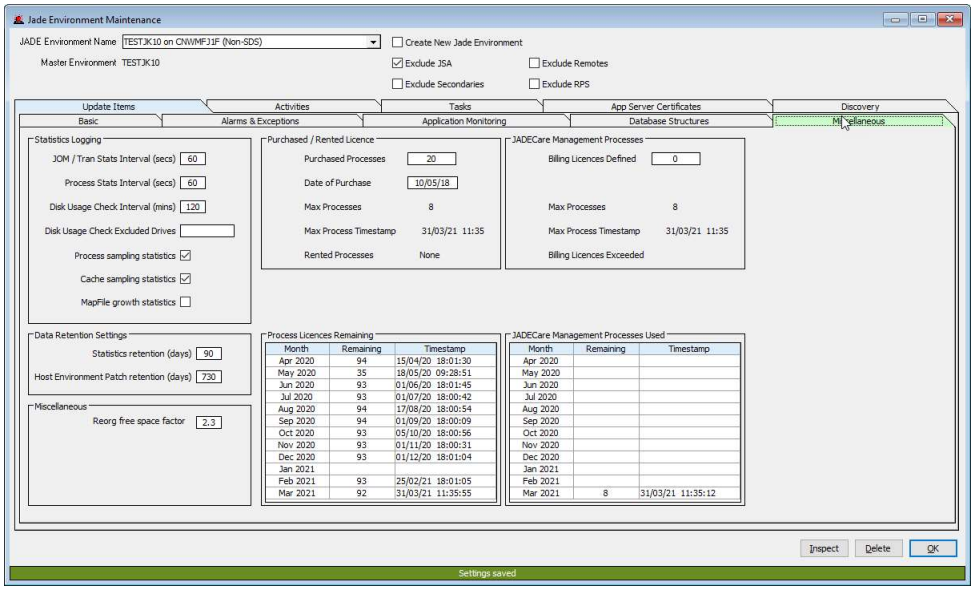
Also from JADE 6.3.07 and CardSchema 3.5.02, the ability to distribute database structures over multiple volumes is supported, as well as the ability to backup the database over multiple volumes.

The specification of the volumes to which a backup is to be directed if other than the standard or pre-deploy volume is specified on this form and is by database structure. Note that any volume specified will be suffixed with the <backup directory>\<env>\server\c_system directory name.

The location and physical relocation of a database structure can be achieved using JADE logic or use of jdbadmin.exe and should be organized with your JADECare Operations representative. Any new volume used should also be added to the Database Volumes Allowed table provided in the Basic sheet so as to avoid alerts advising of structures using invalid volumes.

The columns regarding encryption will only be visible when the Jade Environment has upgraded to CardSchema 3.9.03 or later.

Miscellaneous sheet



Statistics Logging

JOM / Tran Stats Interval (secs)	Defines the transaction sampling frequency in seconds. If zero, then no transaction sampling is done.
Process Stats Interval (secs)	Define the process sampling frequency in seconds
Disk Usage Check Interval (min)	Define the frequency of disk usage checks. Note that this starts a separate application KarmaDiskUsageChecker per host environment where the interval is not zero.
Process Sampling Statistics	Indicates whether or not process sampling statistics are to be collected for this environment
Cache Sampling Statistics	Indicates whether or not node sampling statistics are required. This requires that the ini file setting has not specified Defeat Node Sampling
Mapfile Growth Statistics	Indicates whether or not the mapfile sizes for the host environment are to be captured.

Data Retention Settings

Statistics Retention (days)	Defines the number of days that statistics will be retained for this host environment

Host Environment History retention (days)	Defines the number of days that history details will be retained for this host environment

Reorg Free Space Factor

This field is only displayed for JSM installations having the tracker module. If drive available space is less than reorg free space factor multiplied by the database size, then a tracker will be created.

The factor value specified is between zero and 99.9. The default value of 2.3 is set for all JADE environments (excluding JSM and JSAs) during the migration to JSM 3.8.10. For new environments, the value needs to be explicitly set in the Jade Environment Maintenance form.

If a value of zero is specified (or the JSM installation does not contain the tracker module), then the check will not be performed.

The check is performed only once per fortnight and is triggered when JSA receives database file information (database file sizes and locations) from the managed environment. I.e. when the environment is started and when the environment is backed up.

For environments that span multiple drives (i.e. have map files with explicit file paths), the check is performed for each drive.

Purchased / Rented Licence

This allows a customer to arrange with JADECare 24 for the purchase of extra licences that may or may not be used. However, if any extra licences are used within the last month then this will be invoiced at a premium rate for the whole of the month.

Purchased Licences	Defines the number of licences that the customer is paying for
Date of Purchase	The date that the licences (plus extra) were purchased. Provides a record for the CSM as to when this occurred.
Max Processes	The maximum number of processes that were used in the month.
Max Process Timestamp	The date/time when the maximum number of processes occurred.
Rented Processes	The number of extra licence processes that the user is not paying for and is to be invoiced for the month concerned.

JADECare Management Processes

This area identifies the number of JADECare managed service processes that have been used to date. These are the applications defined in CardSchema that are a requirement for any Jade Environment that is managed by the JADECare Managed Service. This includes applications such as:

- KCCMGr – the communications interface to JSA
- KCLockChecks – executes when extended lock checks are defined in the ini file
- KCSDSChecks – required when performing SDS checking for journal lag and connection status.
- KCStatsCollector – when transaction statistics are being collected
- KCVerifyJournal – used to verify the journal integrity whenever a journal is released
- KCOnLineBackup – the master process when perform an online of 'offline' backup
- KCOnLineBackupWorker – from 1 to <n> workers which perform the database backup
- KCOnLinePartitionWorker – from 1 to <n> workers which perform the backup of database partitions if partitions are used

The derivation of this is obtained as each application registers / de-registers in JSM.

Billing Licences Defined	Defines the number of 'special' licences included in the licencing system to allow for JADECare Managed Service. These are addition process licences provided that the customer does not pay for.
Max Processes	The maximum number of JADECare processes that were used to-date this month.
Max Process Timestamp	The date/time when this maximum number of JADECare processes occurred.
Billing Licences Exceeded	The number of JADECare processes that exceed the 'Billing Licences Defined'. Only displayed if the 'Billing Licences Defined' is other than zero.

Process Licences Remaining

The table shows the current 12 month window of the minimum process licences.remaining and the date/time of when that minmum value was captured.

Gaps could occur in the display for the following reasons:

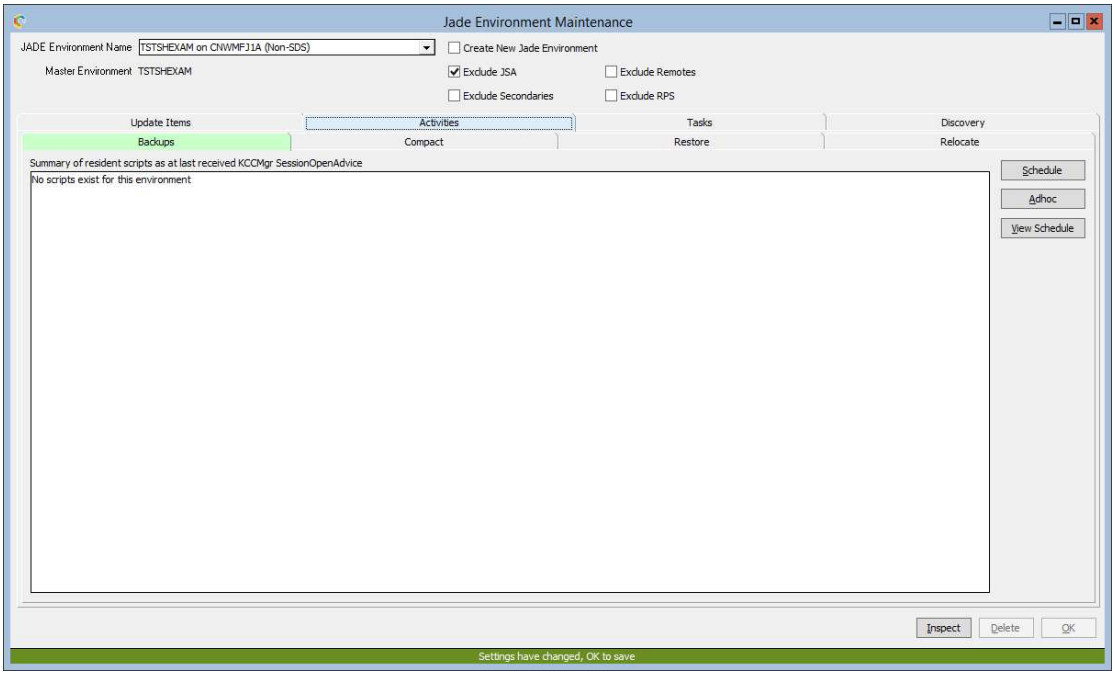
- The information is populated from the transaction statistics retained for the environment athe time of JSM 4.0.03 migration from 4.0.02. As the transaction statisics retained may not cover the full 12 month window, there may be gaps.
- Transaction statistics gathering may be turned off for the environment for a period of time.
- Transaction statistics gathering may never have been turned on and so no data is captured.
- The environment may have been newly created ad so transaction statistics will only start from the date that transaction statistics first start being gathered.

JADECare Management Processes Used

The table shows the current 12 month window of the JADECare management applications maximum processes used and the date/time of when that maximum value was captured.

The derivation of this is obtained as each application registers / de-registers in JSM.

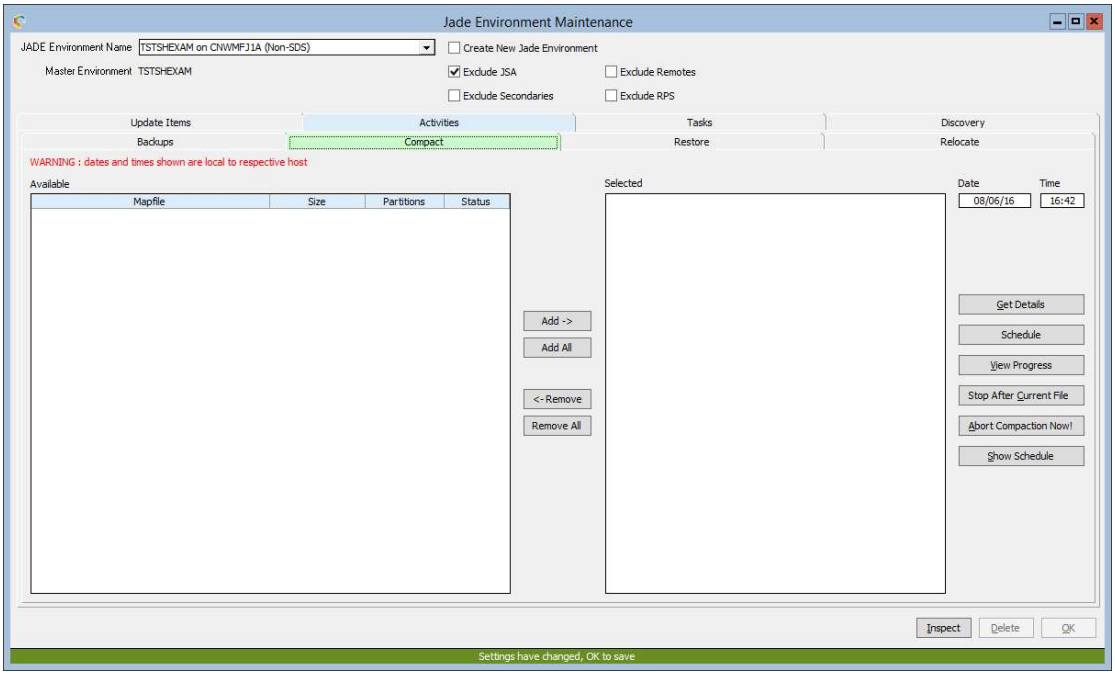
Backups Sheet



Buttons which enable access to the backup schedule, adhoc backup initiation and script setup are presented on this sheet.

For details regarding the buttons ‘Backup Schedule’ and ‘Backup Adhoc’ refer to **Database Backup** user document. This document provides a complete description of the backup facilities offered starting with the JADECare Systems Manager 2.7 release

Compact Sheet



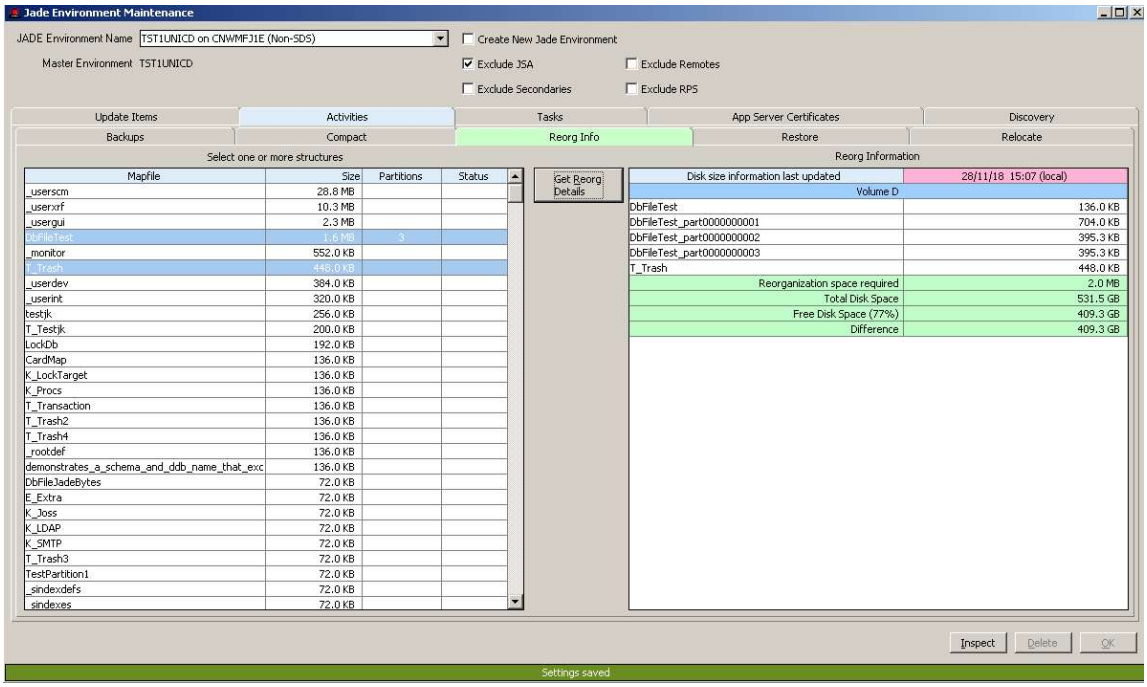
This sheet is used to start an online compaction on selected structures and must be done at the mapfile level. For a mapfile to be compacted, all partitions associated with the mapfile must be online. As can be seen in the above example, there are two mapfiles above which have at least one partition each that is **offline**, and therefore these two mapfiles cannot be compacted until all subordinate structures are **online**.

The online compaction requires that the ini file settings for **[JadeReorg] ReorgWorkDirectory** and **ReorgBackupDirectory** be set to <default>. This ensures that the compaction is done on the same volume on which the database structure resides and avoids any downtime, and exceptions, incurred by a copy to/from alternate volumes.

Note that where a mapfile has one or more partitions then if ini file setting **[JadeReorg] ReorgWorkerThreads** is greater than 1, then these files can be compacted concurrently.

Reorg Info Sheet

The following sheet is intended to provide information as to whether there is enough free space on the database volume(s) to perform a successful deployment where a reorganization will be performed.



This sheet loads a table with known mapfiles (same data as shown in compact sheet but sourced solely from data in JSM).

One or more of these mapfiles can be selected and when the 'Get Reorg Details' button is clicked this will:

- work out the disk space used by volume for the mapfiles (and associated partitions, UDR files etc)
- if server system statistics includes disk info and these exist in JSM then this info will be used.
- The above information is displayed by volume listing:
    - the date/time the disk information being used was last updated
    - volume name
    - database structures defined for the volume
    - the disk space used for the structures on this volume
    - the total disk space for the volume
    - the free space for the volume
    - the difference between structure space used and free space identified

    The items above are repeated if more than one volume is used for the database.

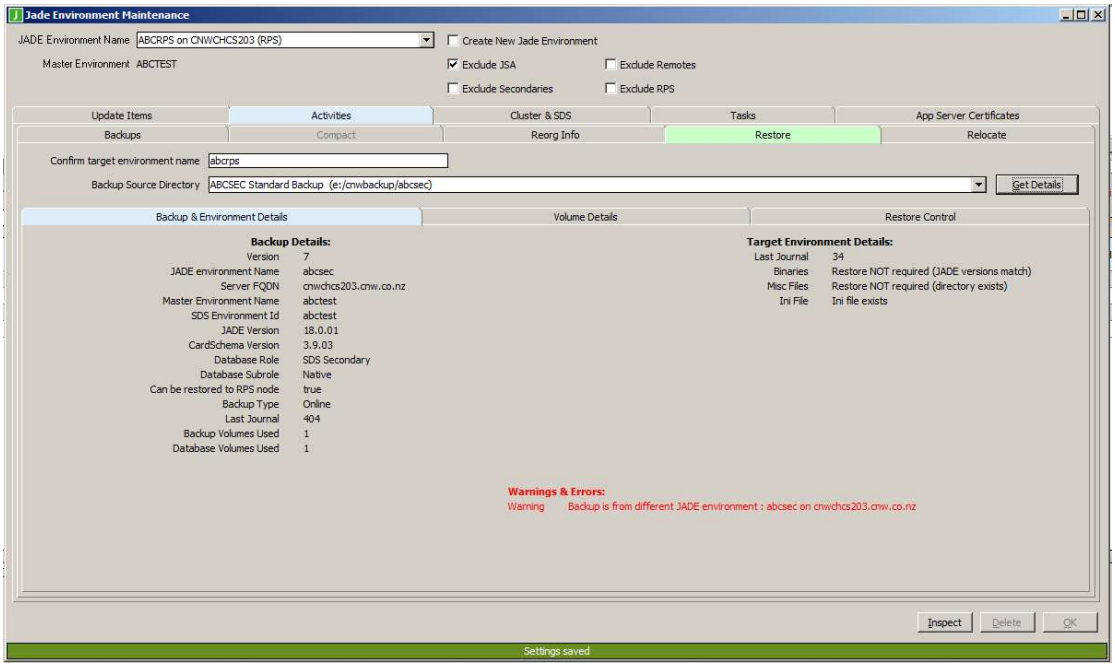
If there is no disk info gathered or retained in JSM for the server the total disk, free space and difference rows will not be shown and a warning dialog will be

issued. The 'Disk size information last updated' date/time cell will display 'Unavailable' and have a highlighted background.

If the disk info data is more than a day old then a warning is issued and the date/time is shown in a highlighted background.

Restore Sheet

The following sheet is shown if the target environment is managed by a JSA 4.0.01 or later.

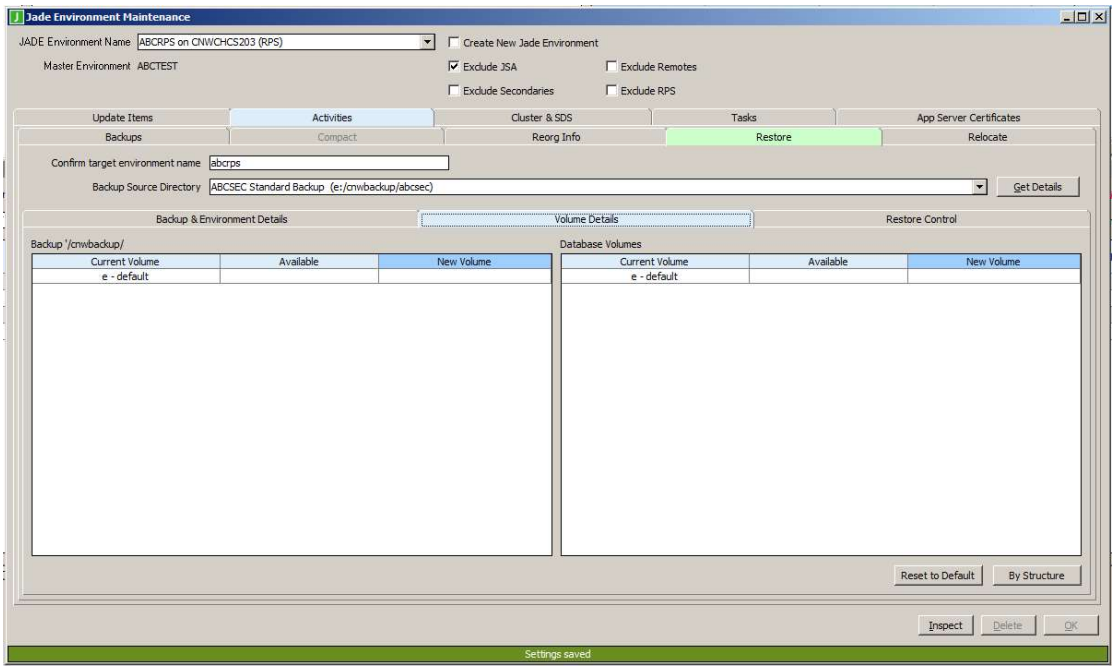


The above display is that after the Get Details button has been clicked and the requested information has been obtained from the backup specified and returned to JSM.

In this case, the form refresh was preceded by one of the pop-up dialogs, see below, that there were 2 database structures that were offline.

The above pop-up dialog is a warning as the host environment has been marked as **Offline Structures Allowed** on the **Managed Environment Backup** form.

The above pop-up dialog is an error as the host environment has not been marked as **Offline Structures Allowed** on the **Managed Environment Backup** form.

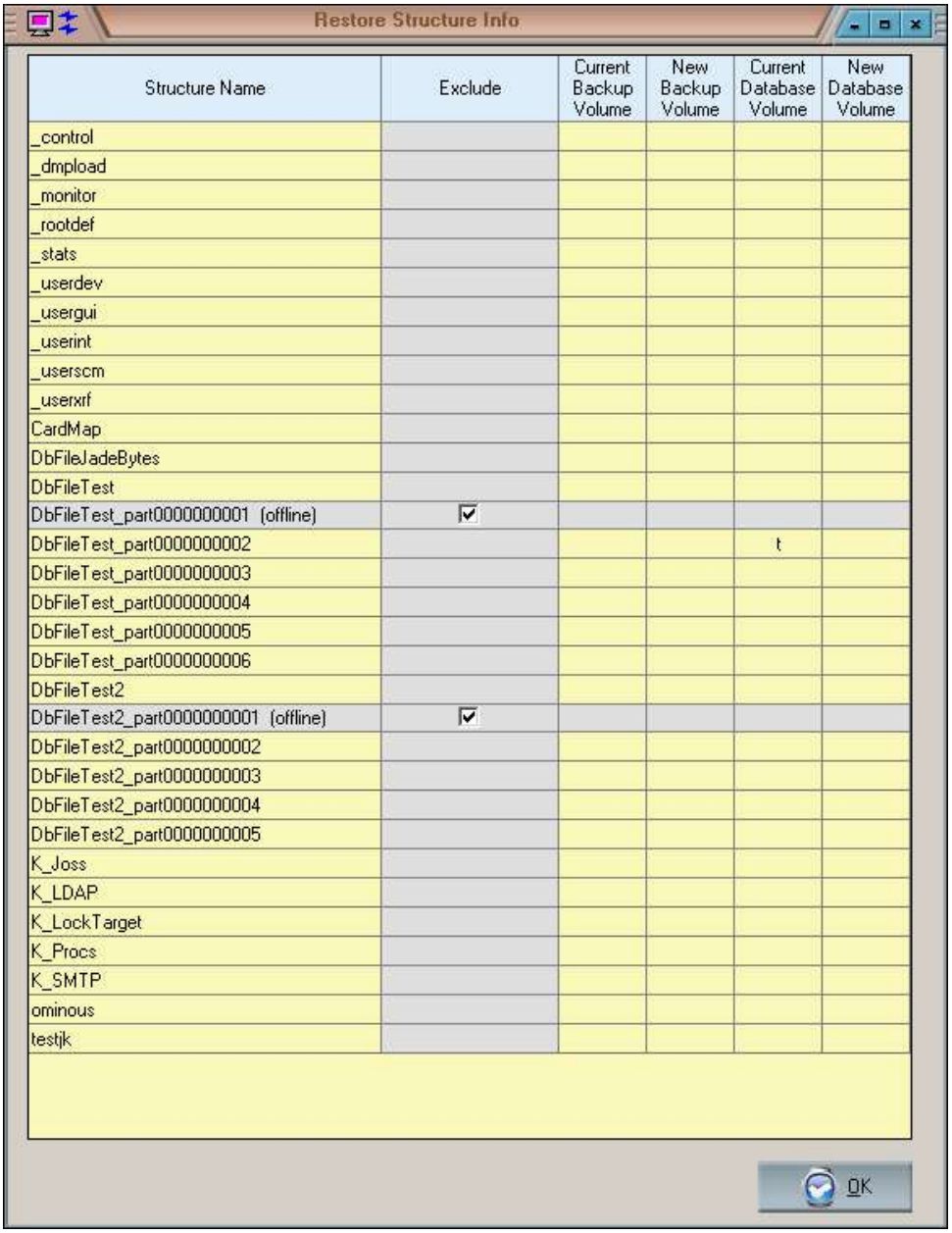


The **Volume Details** sheet allows for the macro management of volume changes for both backup and database volumes should these be necessary.

Where the backup and/or database tables above show more than the single default volume, the corresponding cell under heading **New Volume** contains a combo list of valid alternative volumes.

Should the volume derived from the backup instance not be a valid volume on the server, then the corresponding cell under heading **Available** will show 'No'.

Once changes have been effected either by this sheet or by individual structures using the **By Structure** button and form **Restore Structure Info**, then this detail can be discarded by using the **Reset to Default** button.



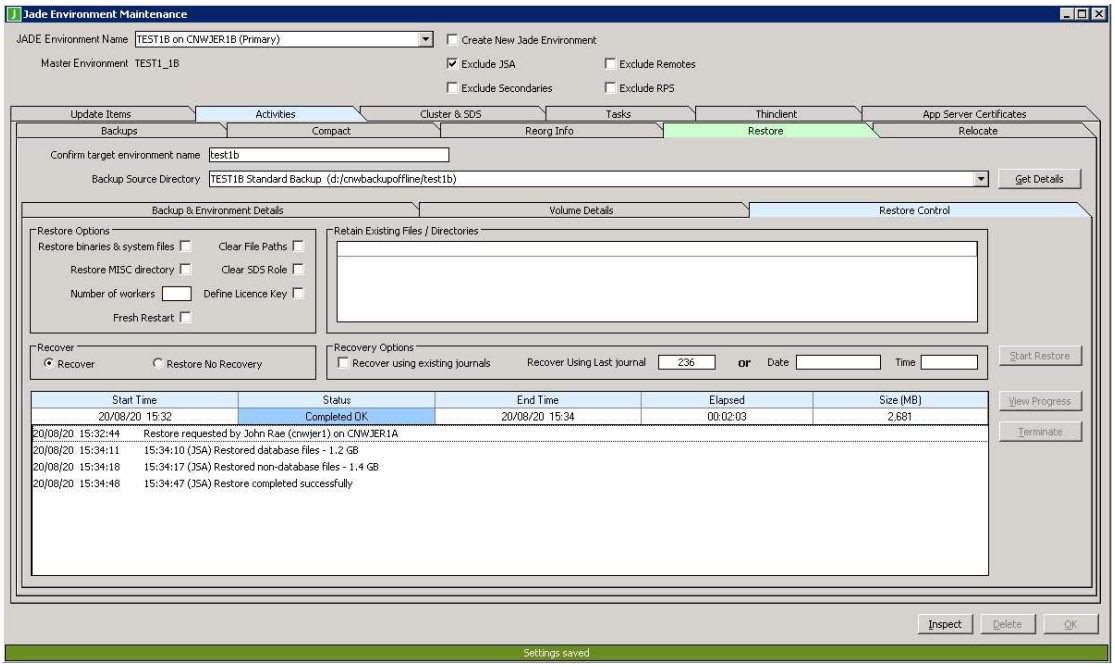
The above form is displayed by clicking the **By Structure** button.

It displays :

- every database structure, mapfile or partition
- whether it is offline or not
- the current backup volume, if different from the default backup location
- a combo box listing valid alternative volumes to the backup volume, if required
- the current database volume, if different from the default database location
- a combo box listing valid alternative volumes to the database volume, if required

When the host environment is configured with JADE 6.3.09 and **Offline Structures Allowed** has been set:

- then it will be possible for the restore to exclude one or more partitions from the restore. Any partitions that are excluded will be marked as **offline**. Note that if this exclude option is used, then there is a possibility that recovery phase using journals may fail if it attempts to access a partition that has been excluded and is marked **offline**.
- [cnwrefreshparamters_<env>_<server>] entries from the backed up ini file are returned and, if specified, are used to exclude all but the required most recent partitions and/or for the volume (and directory <vol>:\<env>\server\c_system) to be specified for all parts of the named mapfile and sub-files (index and partitions)



The **Restore Control** sheet of the **Restore** function provides for key attributes regarding the restore as well as the initiation, termination and viewing the progress of the restore.

Restore Options:

Restore binaries & system files	This check box affects whether or not the binaries and system files (*.bin) are restored from the backup.
Restore MISC directory	This check box affects whether or not the files in the server\c_misc directory are restored from the backup. **NOTE:** As from release 3.8.08, if a directory <env root>\server\c_misc\apps exists, then this will ALWAYS be restored regardless of the setting of option 'Restore MISC directory'.
Number of workers	This value specified how many restore workers are to be running when restoring database structures. This value is prefilled in with the value defined for the number of workers for an offline backup.

Clear File Paths	Where database structures for the backed up environment exist over multiple directories and volumes, this can be reset so that all database volumes reside in the same directory as the control file (_control.dat).
Clear SDS Role	When restoring from a SDS host environment, the SDS database role can be cleared by selecting this option.
Define Licence Key	This will cause a dialog to pop-up once the **Start Restore** button is clicked allowing for the entry and validation of licence key details.
Fresh Restart	This setting only applies when a subsequent restore is being performed after a prior restore that failed in the non-database file copy phase and the database was subsequently opened (thus inadvertently starting a new history independent of the backup journals). Setting Fresh Restart causes the restore to always start at the beginning (db file copy phase) so that roll forward recovery can be performed with the intended journals. If Fresh Restart is not set AND the previous restore had failed in the non-database copy phase then the restore will resume at the non-database file copy phase.
Restore No Recovery	When this option is selected, the Recovery Options group box is not enabled and the operator assumes responsibility for restoring journals into place and performing recovery as required.

Recover To:

Recover using existing journals	When this check box is checked the restore will expect and use the required journals in the host environment **JournalRootDirectory** otherwise the expected journals will be retrieved from the backup instance..
Last Journal	The number of the last journal to be recovered to. At least one option of **Last Journal** or **Date/Time** is required for non-SDS or SDS Primary.
Date, Time	Date in format dd/mm/yy and time in format hh:mm:ss being the date/time up to which the database is to be restored.

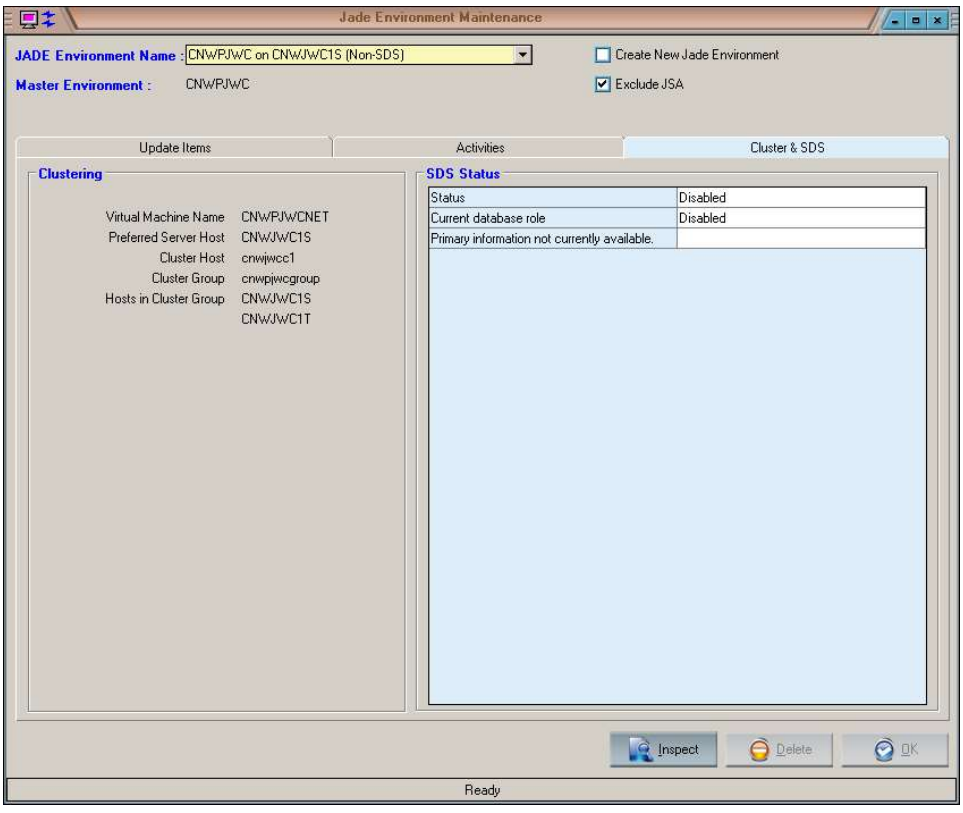
It is expected that the journals required for this type of recovery are available in the location specified by **Restore using existing journals**.

Notes Regarding Usage of 'Restore No Recovery'

1) When 'Restore No Recovery' is specified, the Recovery Options Group Box is not enabled and the restore is performed with no recovery. If the backup used is an online backup then recovery must be manually performed (i.e. using jdbutilb recover command).
   Particular care should be taken when restoring an SDS primary using this method (since the journal current at the time of backup completion is required and you may want to recover to a point in time that enables existing secondaries to catch up).

   When restoring an offline backup to a SDS Primary, a manual recovery step is still required since the backup journal is required for a SDS Primary and the backup is achieved as an online quiesced backup and the journal advances during the backup.
   When restoring with no recovery to a standalone environment from an offline backup, the backup journal is only required if you wish to perform a roll forward recovery. Putting the backup journal in place means that recovery is necessary (again, because a quiesced backup is being restored).
2) The usual post recovery restore steps of performing an abort reorg and unversion are skipped when restoring an online backup with no recovery (since this opens the database and results in automatic backup recovery).
3) The operator manually performing the recovery is responsible for Restoring journals and clearing the database role, if required.

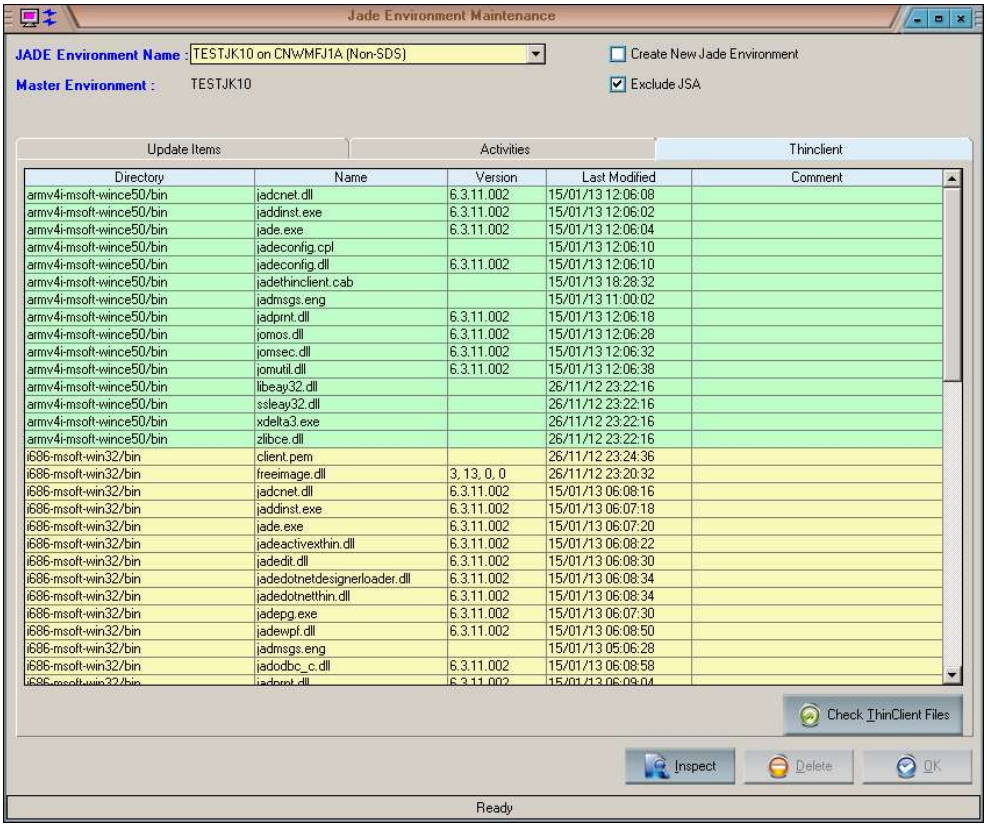
Cluster & SDS sheet



This sheet is visible only for those host environments that are defined as SDS Primary, SDS Secondary Native or SDS Secondary Relational. The information displayed on the sheet is updated periodically (based on the specified ping interval) when pinging is enabled on the **Alarms and Exceptions Sheet** and when the environment changes to active or inactive

When a host environment (database or remote) is defined as part of a cluster resource, the pertinent cluster details and associations are shown here.

Thinclient Sheet

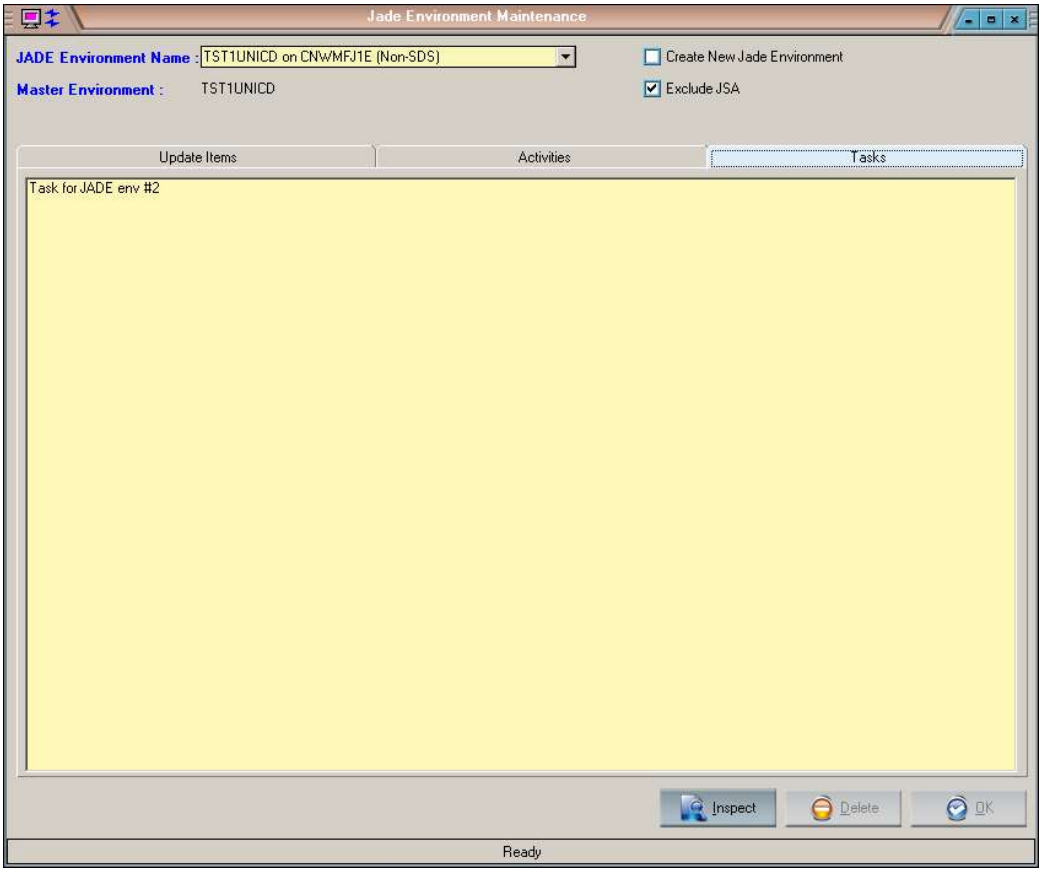


This sheet is only visible when the master environment has been defined with deployment regime 3.

It is available for both database and remote host environments and lists the thinclient file details as compared to the master environment definition. The 'Comment' column will be highlighted in Pink when an anomaly exists with the thinclient files declared for the master environment, the comment will identify what that anomaly is.

The button 'Check ThinClient Files' can be used to refresh the above details from the specific host environment being viewed but may take a little time to collect from JSA.

Task Sheet



This sheet lists all of the task defined for this JADE Environment.

Double-clicking on a row containing a task will bring up the Task Maintenance form for that task.

Setting up for SDS Environments

JADECare Start 3.1 in conjunction with JSM 3.1 allows you to configure host environments with names that are different from the master environment name. For example:

Master environment id	TEST3161
Primary host environment name	TEST3161A
Secondary host environment name	TEST3161B
RPS host environment name	TEST3161R

Only one SDS primary for a given master environment can reside on a particular host, however by using unique names as show in the example above, it is possible to run any combination of primary and one or more secondaries on a single host. Association of primaries with secondaries and their master environment is achieved using the following two new ini file options in the managed host environment's initialization file:

```
[AppEnvironment]
MasterEnvironmentId=TEST3161
SDSEnvironmentId=TEST31
```

SDSEnvironmentId is an arbitrary identifier used to uniquely group and identify all databases that belong to a given SDS primary. Only one primary can use a particular SDS environment id. The maximum length of this identifier is 30 bytes.

MasterEnvironmentId is the name of the master environment to which all the managed environments belong. This is a string of max length 9 bytes, and is defined on the Master Environment Maintenance screen in JSM. If the MasterEnvironmentId is not specified, it will default to the managed environment name.

Jade environment names are derived as in earlier releases of JSM, from the top level directory name of the environment concerned. This will be:

(Windows)	<driveLetter>:/<environment name>
(Linux)	/opt/<environment name>

For more information about setting up your SDS environment, see subject **Setting Up for Auto-Updating Deployments**, later in this document.

JADE Environment Definition

JadeEnvironment objects are normally automatically created (discovered) by JSM when the managed JADE Environment connects to JSA which JSA will forward to JSM. JADE Environments may also be manually defined.

SSL Certificates

Where an Application Server (jadapp) is executing for a JADE Environment and it executes one or more applications that invokes CardSchema method 'app.cnActivateKarmaControl', then any ini file SLL certificate details for the Application Server will be sent to JSM creating/updating an SSL Certificate object.

These SSL certificates can be viewed via:

- Jade Environment Maintenance form's sheet App Server Certifcates
- Certificate Summary form

These SSL Certificates are subject to the certificate expiry check (certificate expiry threshold period defined in the System Global Options) in the same way as the IIS Certificates are.

Global Parameter File

The global parameters consist of multiple separate files that are to be found under the directory c:\Cardinal\etc wherever JSM or JSA, is executing.

Note that as at release 3.8.04 that the global parameter information below will be found in two files for each:

- Suffix .txt – JADECare key format
- Suffix .json – Details are provided in JSON format

These files are titled:

glbhost
contains information global to server hosts.

glbenv
contains information specific to JADE managed environments

glbnet
contains information specific to .Net Services

glbodbc
contains information specific to ODBC monitored environments

glbvar
contains information that is globally required and not related to hosts or environments

glbweb
contains information specific to website applications

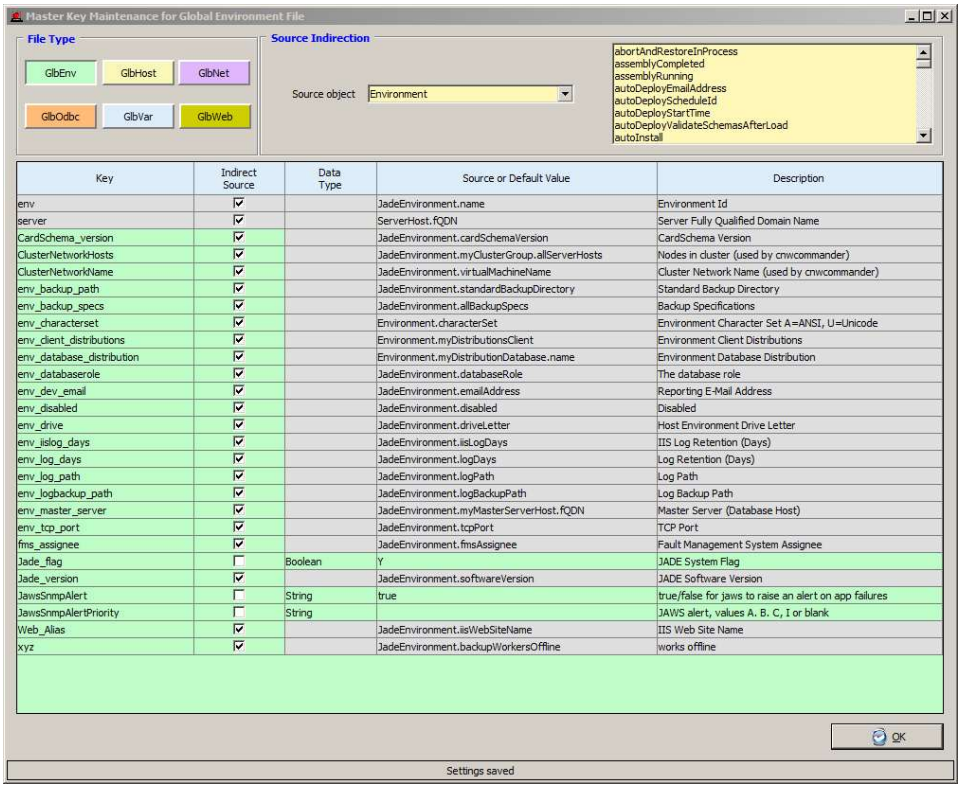
The creation of the information provided in the global parameter files is obtained indirectly from the data maintained internally by JSM on each:

- .Net Service
- Server Host
- Managed Environment
- Master Environment
- ODBC Managed and/or Monitored Database
- Website Application

or explicitly by the entry of user data within the global parameter definition screens.

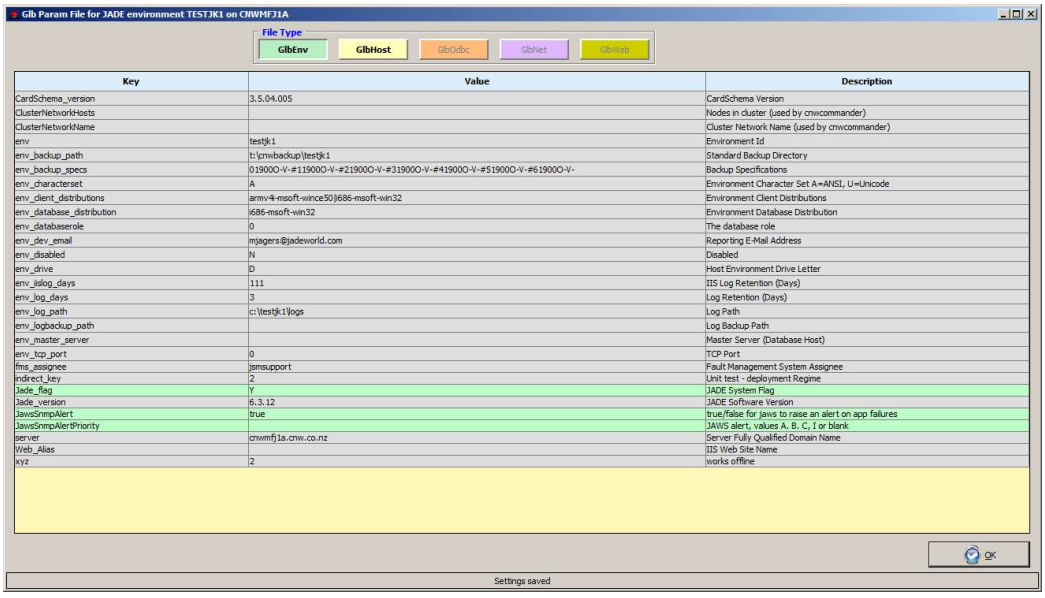
The above files are accessed by various scripts and tools developed by JADECare Systems Group and JADECare 24 Operations. The global parameter files are derived from data held in JSM and are distributed to every server where JSA is executing.

The details that form the global parameters are defined via the following screens:

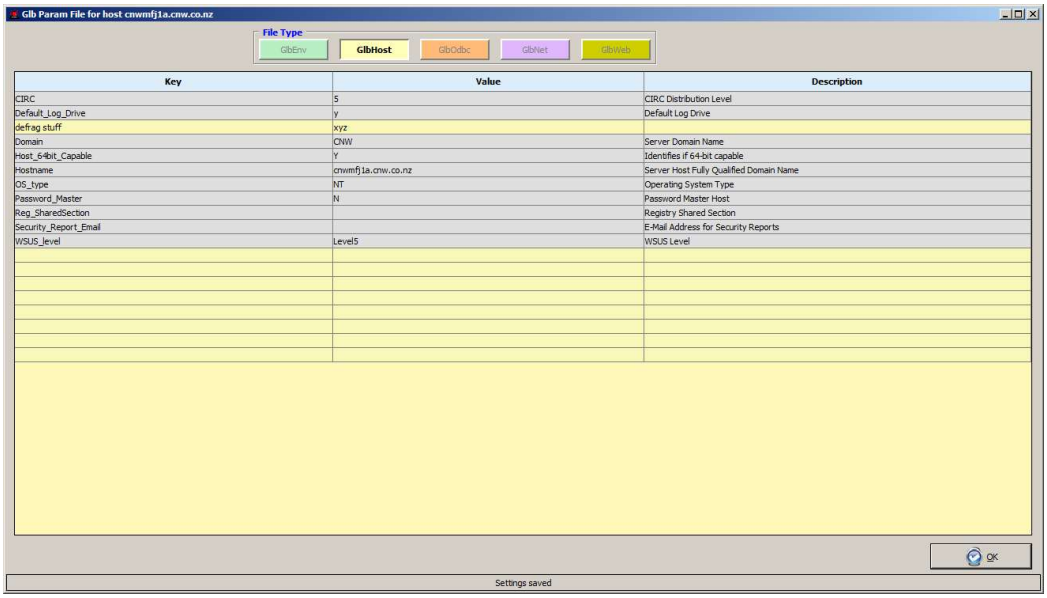


Once the master definition of what is to be contained in the global parameters has been set up (defaults are provided by JADECare Systems Group), the actual information that will be derived can be viewed :

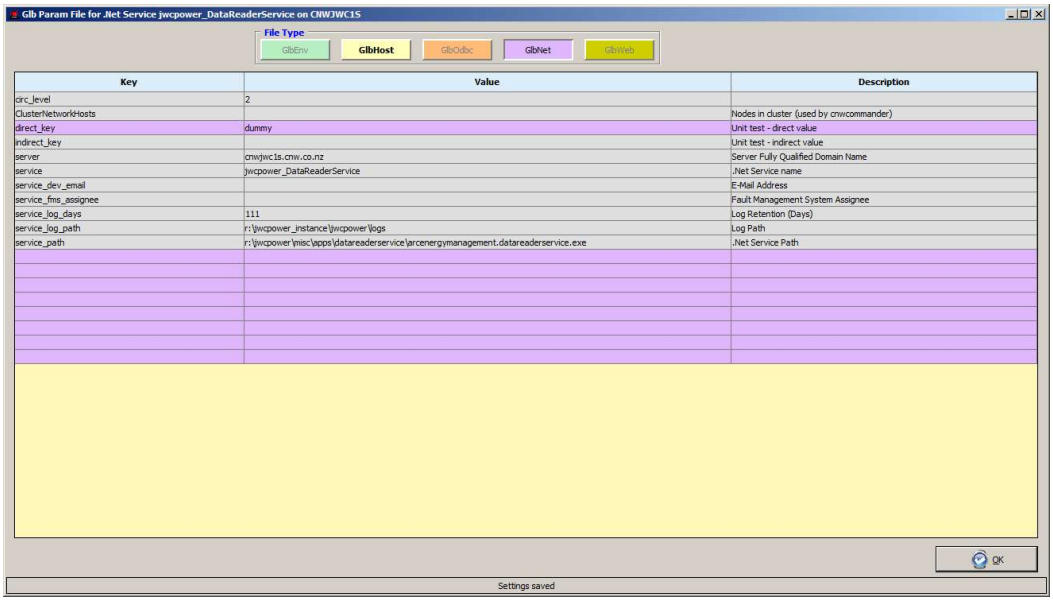
for each Jade environment via the **Jade Environment** screen button **Global Parameters**:



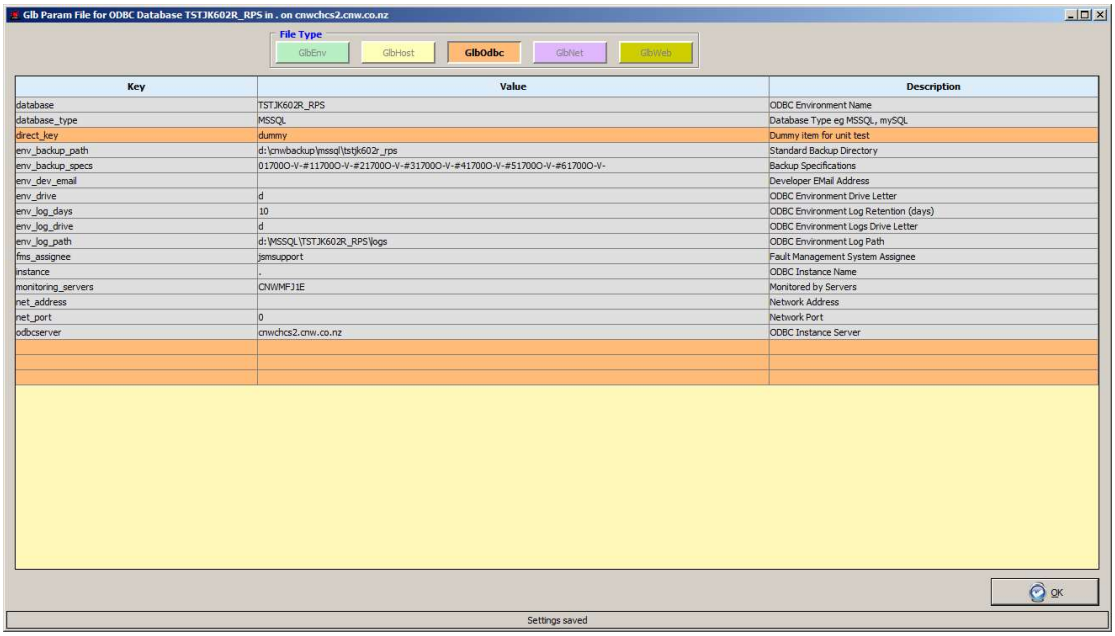
for each host via the **Hosts** screen button **Global Parameters**:


Glb Param File for host cnwmfj1a.cnw.co.nz
File Type
GlbEnv
GlbHost
GlbOdbc
GlbNet
GlbWeb
Key
Value
Description
CIRC
5
CIRC Distribution Level
Default_Log_Drive
y
Default Log Drive
defrag stuff
xyz
Domain
CNW
Server Domain Name
Host_64bit_Capable
Y
Identifies if 64-bit capable
Hostname
cnwmfj1a.cnw.co.nz
Server Host Fully Qualified Domain Name
OS_type
NT
Operating System Type
Password_Master
N
Password Master Host
Reg_SharedSection
Registry Shared Section
Security_Report_Email
E-Mail Address for Security Reports
WSUS_level
Level5
WSUS Level
OK
Settings saved

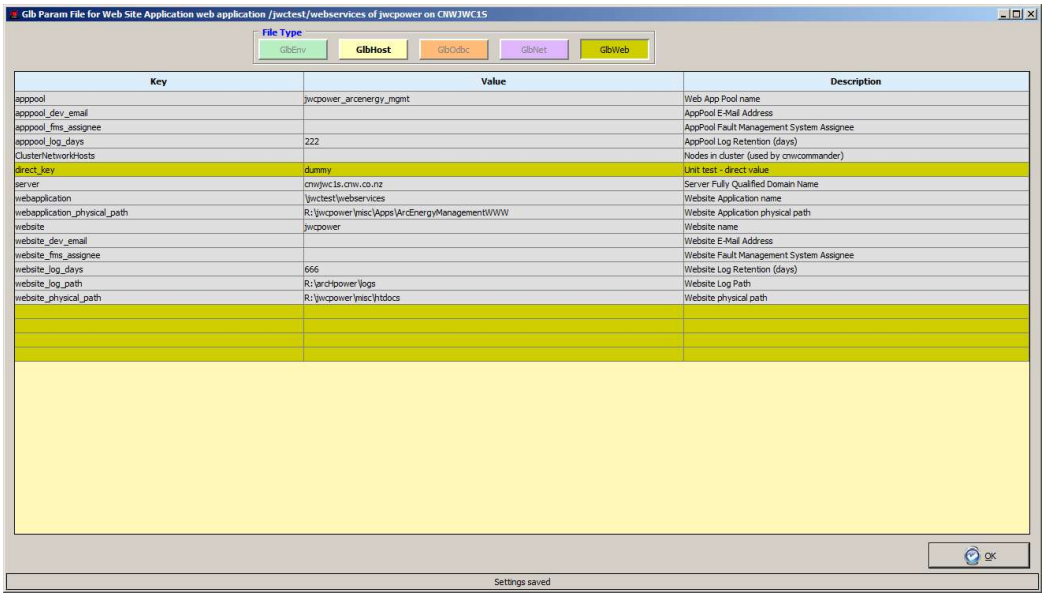
for each .Net Service via the menu item **Global Parameter Maintenance** on the **.Net Service Tree View** form:



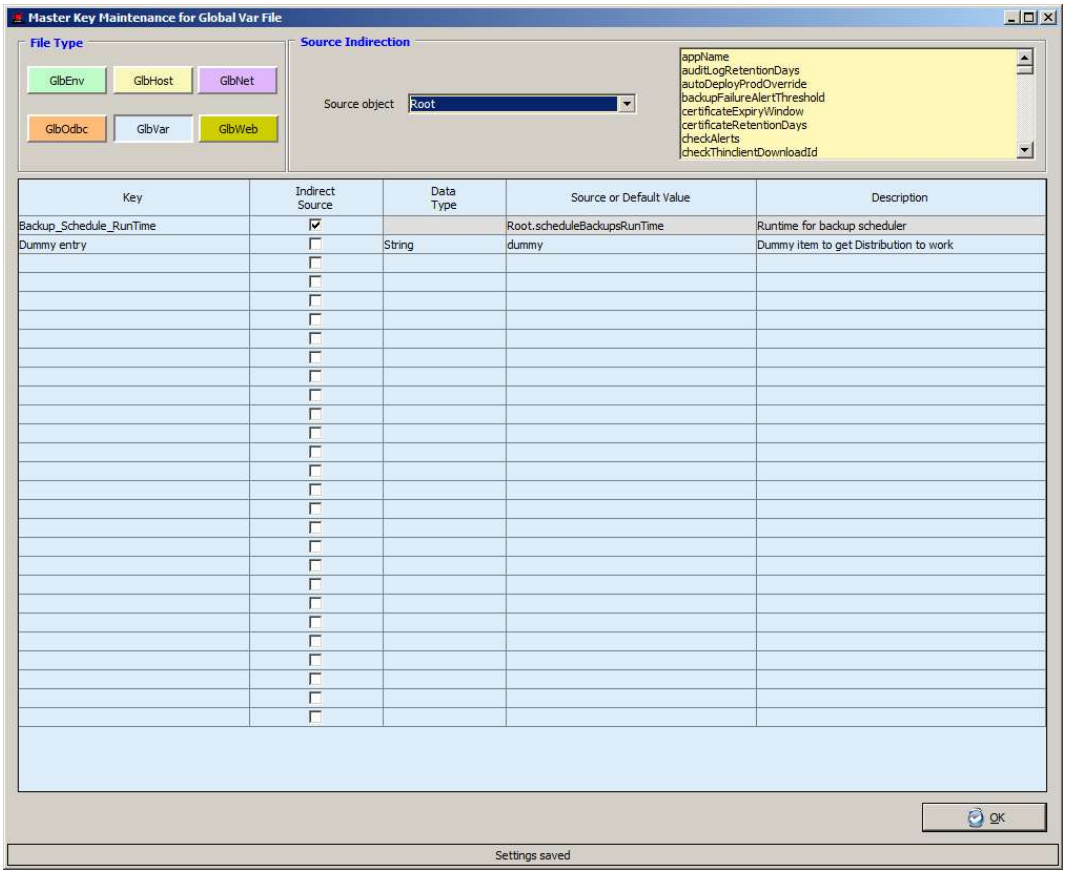
for each ODBC Database via either the **ODBC Maintenance** form using menu option **Global Parameters for this ODBC Database** or **System View** form by right clicking on an ODBC Database entry and selecting menu option **ODBC Environment Global Parameters:**



for each website application via the menu item **'Global Parameter Maintenance'** on either the **Website Tree View** or **AppPool Tree View** forms:



for the global information via the **Master Global Parameter File** and clicking on the GlbVar button.



GlbEnv.txt and GlbODBC.txt details

env_backup_specs

The backup specifications are defined as a string with each backup specification being delimited by the # character.

Format for each backup specification is :

dhhmmtcvz

where:

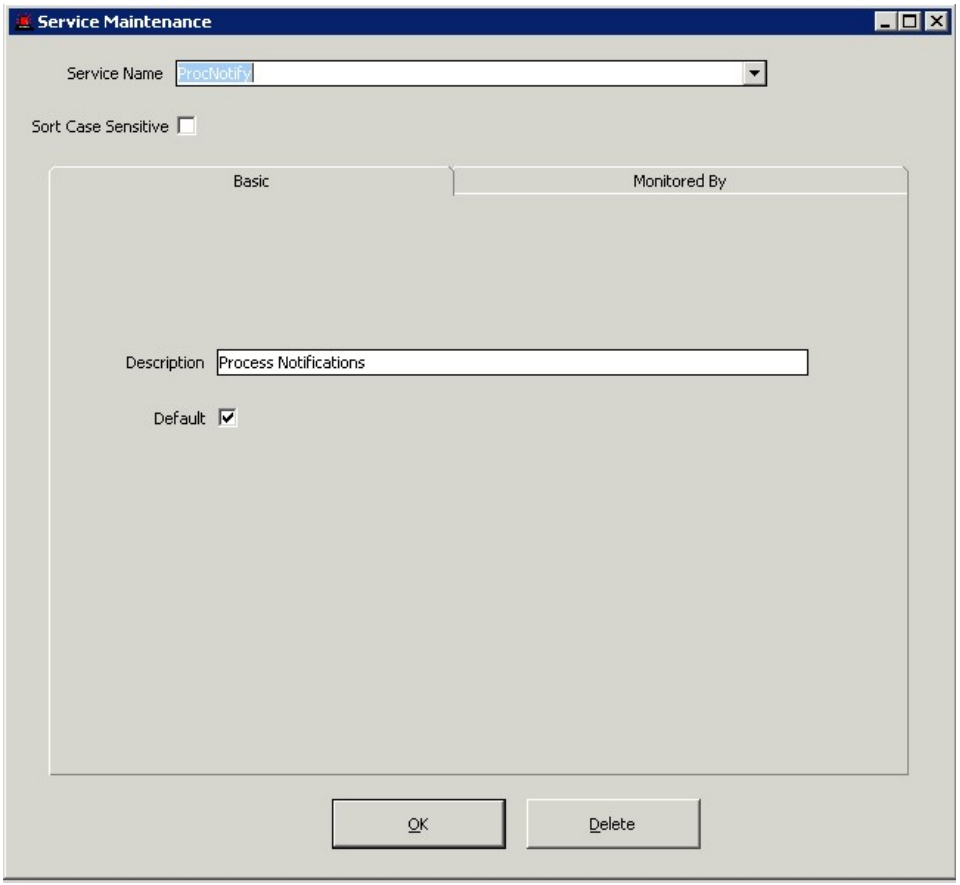
- d the day of the week with 0 = Sun and 6 = Sat
- hhmm the time of day
- t offline (X) or online (O)
- c compression (C) or not (-)
- v this will always be true
- z compression requested (Z) or not (-)

Example:

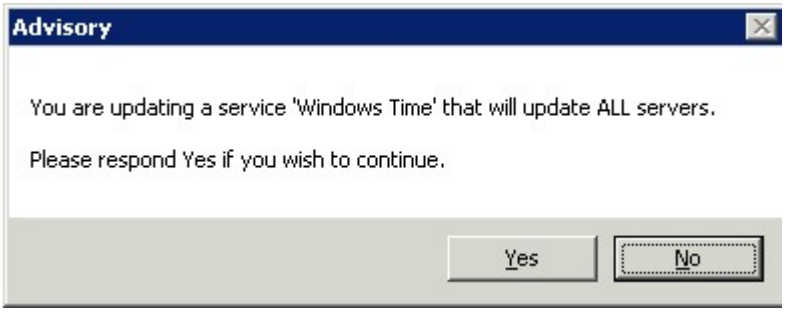
02300X-V-#12300O-V-#22300O-V-#32300X-V-#42300O-V-#52300O-V-

Services Definition

The following form can be used to define or delete Windows services that can be monitored.



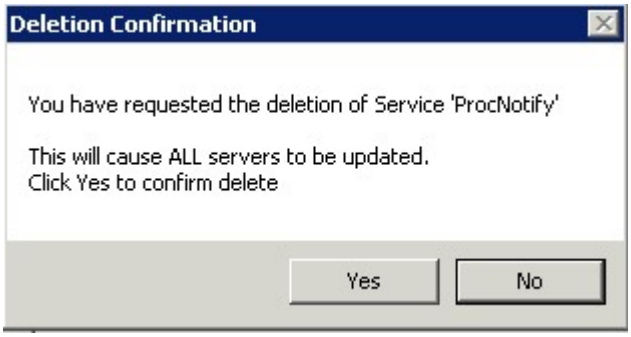
If the 'Default' check box is checked and it wasn't previously then a dialog , as follows, is issued to verify that all servers are to be updated.



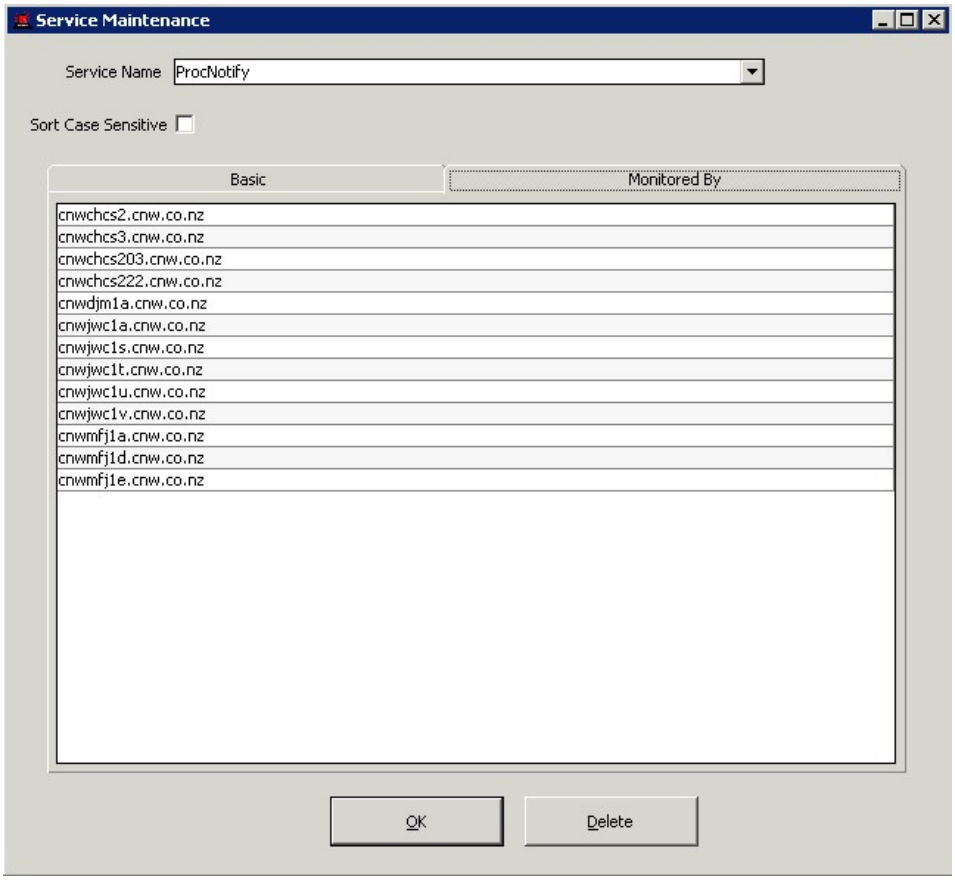
If the 'Defaaut' check box is checked, then this service is added to the default list of services to be monitored and will cause update to all existing server hosts and will update each JSA to automatically monitor this new service.

Any new server hosts added to JSM will automatically have these services monitored by default.

When the 'Delete' button is clicked, then a confirmation dialog, as follows, is issued and will indicate the scope of the change (the number of servers affected):

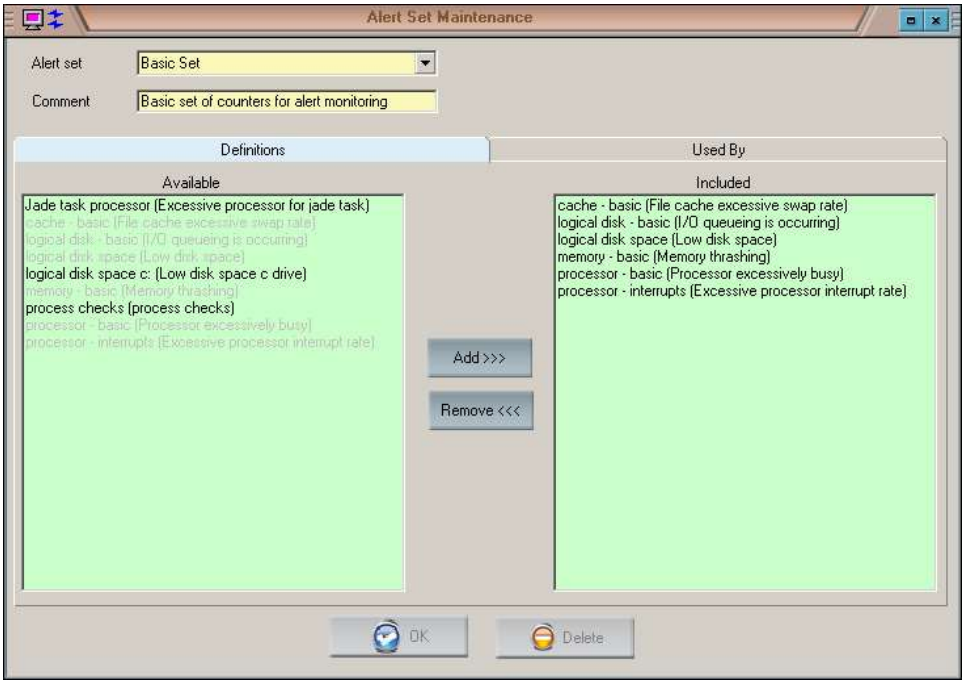


When an existing Service is displayed then the servers that have this service in the list of monitored services will be displayed. Double-clicking on the server row will displayed the ServerHost Maintenance form.



Alert Definitions

Alert Set



An alert set is built from one or more performance counter threshold items (see **Definition on an Alert Set** below). This set can be named and be assigned to specific hosts. Only one alert set can be assigned to a given host at any time.

An alert set can be assigned and the activation of the alert checking set on or off as desired.

If alert checking is already turned on for a host, then changes to the alert set can be forced out to the by using the Cycle option, this avoids having to select Off and click OK then selecting On and clicking OK.

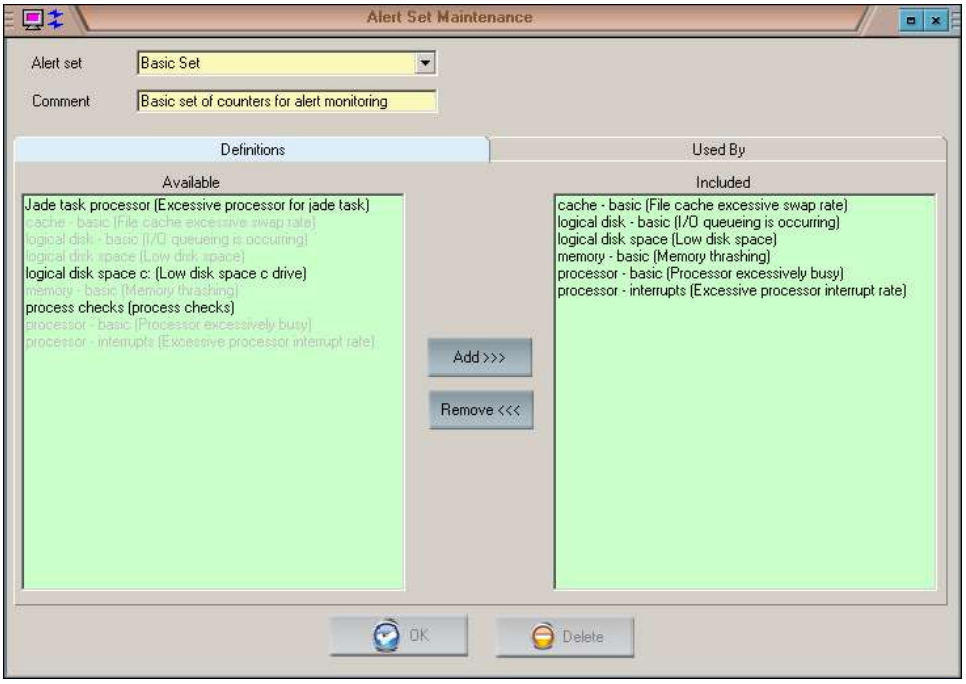
Any changes to the alert set or its status is communicated immediately to the appropriate JSA when OK is clicked.

The **Used By** sheet provides information as to which servers have this

Definition of an Alert Set

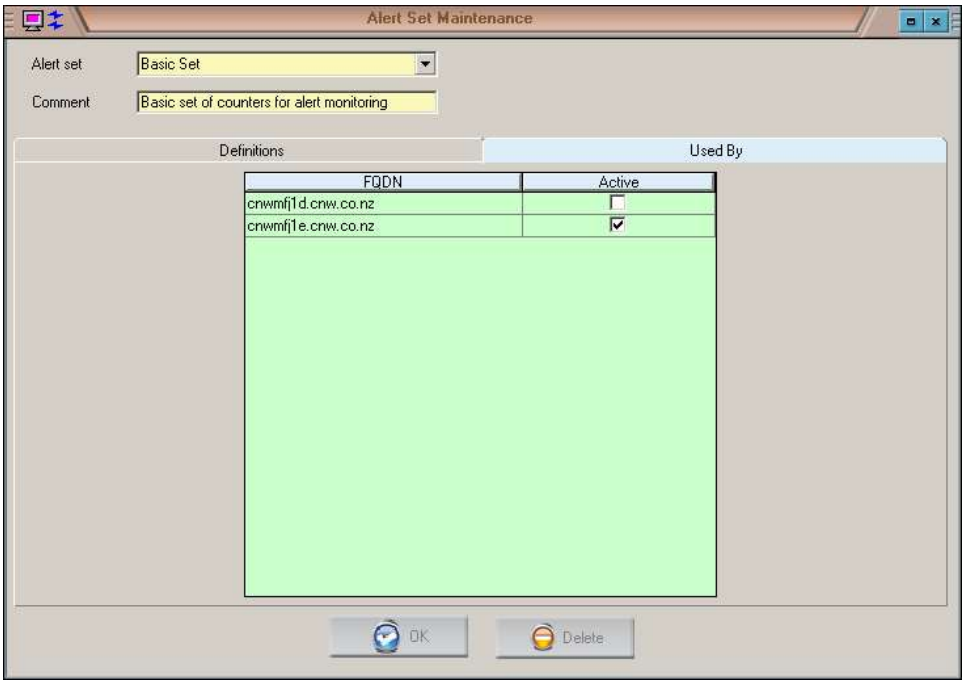
An alert set can be named and the performance items required selected. Existing alert sets can be modified or deleted (except for the **Basic Set** shown).

The left selection box shows those items available for selection (those already selected are shown in light grey, those not yet selected are shown in black).

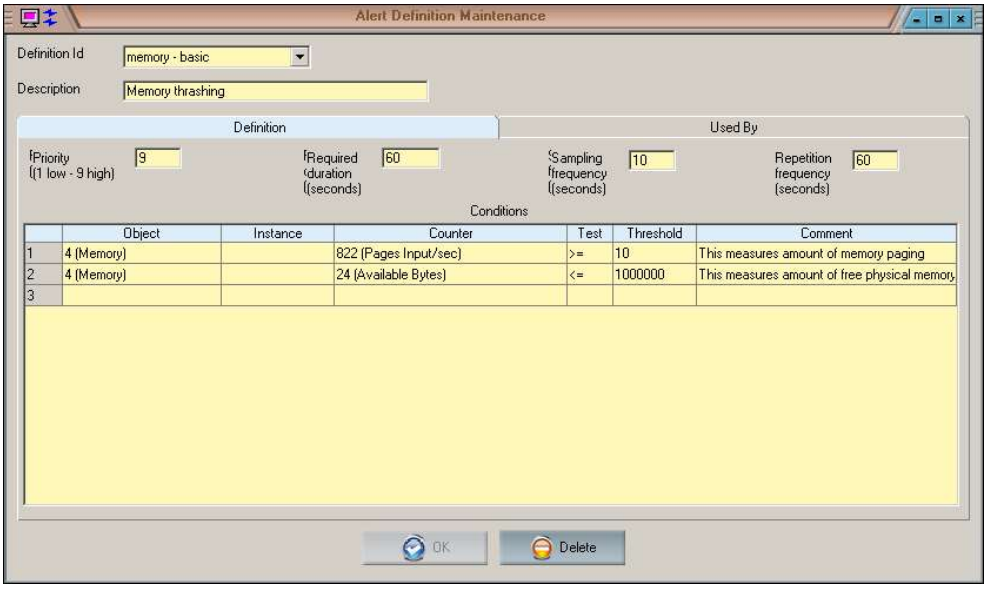


Usage of an Alert Set

The form **Alert Set Maintenance** form sheet **Used By**, shows the server hosts where the alert set has been defined and if it is active.



Definition of Performance Items



One or more of the performance counters / items provided by windows can be defined and thresholds set with the Performance entity being named. These performance entities can then be added to one or more alert sets as required for performance checking on managed server hosts.

Definition Id	the unique name of this performance entity.
Description	a description for this performance entity
Priority	minimum 1, maximum 9.
Required duration	time in seconds
Sampling frequency	time in seconds
Repetition frequency	time in seconds

Conditions

Rows can be added or deleted by right-clicking in the table. Multiple performance counters and tests can be grouped together in this manner to create a performance item.

Object	click on this cell and a listbox of possible performance objects is available for selection
Instance	Leave empty for all instances or identify the specific instance if this is valid
Counter	click on this cell to be given a selection of counters applicable to the performance object selected in cell 1 of this row
Test	possible operators that can be used are provided in a list box (> greater than, >= greater or equal, < less than, <= less than or equal)
Threshold	the value the performance counter is to be tested for.
Comment	free format comment if required (optional)

Ignore Logical Disk Volumes from Performance Alert Checks

With the 4.1.01 release, it is possible to have disk volumes ignored by the performance checks. This is achieved by including the literal (case-insensitive) '_JSMIgnore' in the volume label.

For example, having a volume labelled 'Page Disk _jsmIgnore' will no longer alert if this volume has less than 6% available when there is an alert definition defined which generates alerts when available disk on a value is less than 6%

Application Backups Setup

Refer to separate user document 'Database Backup".

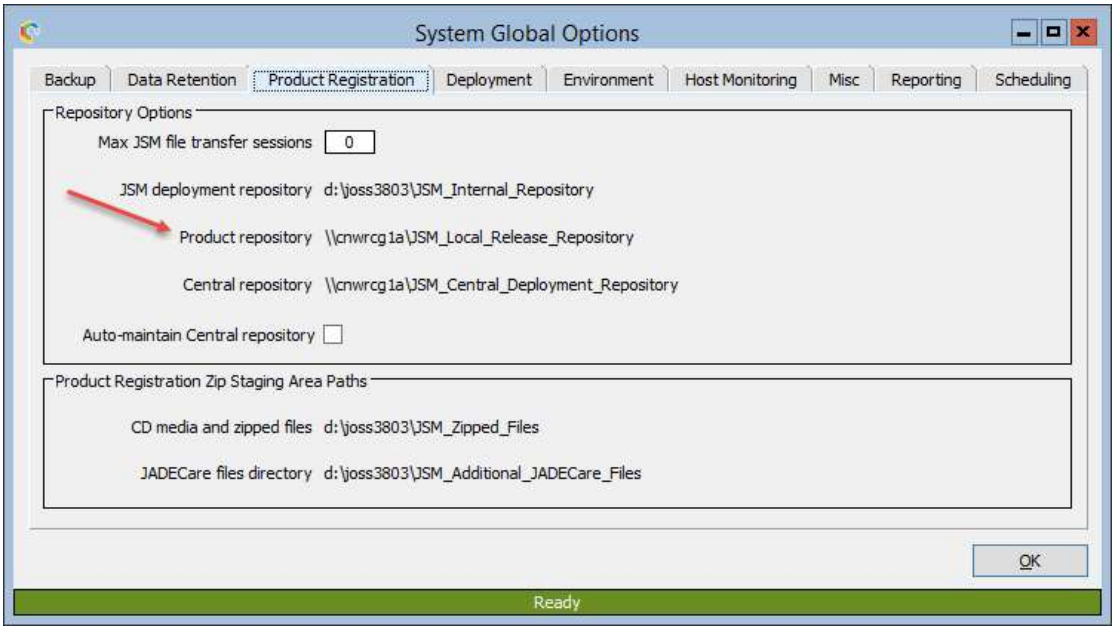
System Configuration Management Setup

Product Master Repository

In order to deploy new JADE and CardSchema releases and hot fixes as part of the automatic deployment features provided in the JADECare Systems Manager (JSM), the Product Master Repository must be defined and maintained by the JSM administrator. The activities involved are described in this section.

Defining the Product Repository Location

The Product Master Repository is defined in the JSM initialization file **[JadeSystemsManager] ProductRepositoryDirectory** and may be viewed via the **Deployment** sheet of the **System Global Options** form in JSM, accessed via menu item **Options | System Global Options.**



Directory Structure

The directory structure of the product master repository is as shown below. This structure will be automatically defined by the BUILD process otherwise must be manually defined.

As new JADE, CardSchema and JADE Master Terminal releases become available, these releases and any hot fixes must be loaded into the appropriate directories prior to registering the release entity via the **Product Details** form.

```
<Product Repository Top Directory>
   CardSchemaReleases
      3.7.04
         i686-msoft-win32
            Base
               a_bin
               a_system
            HotFixes
               a_bin
               a_system
      3.7.05
         i686-msoft-win32
            Base
               a_bin
               a_system
            HotFixes
               a_bin
               a_system
   JadeReleases
      7.0.10
         i686-msoft-win32
            Base
               a_bin
               a_system
               a_sys_p
               a_deploy
               a_dep_p
               a_debug
               a_thinbin
            HotFixes
               005
                  a_bin
                  a_thinbin
               006
                  a_bin
                  a_thinbin
         x64-msoft-win64
            Base
               a_bin
               a_lib
               a_system
               a_sys_p
               a_deploy
               a_dep_p
               a_debug
               a_thinbin
               a_thinlib
            HotFixes
               005
                  a_bin
                  a_lib
                  a_thinbin
                  a_thinlib
               006
                  a_bin
                  a_lib
                  a_thinbin
                  a_thinlib
```

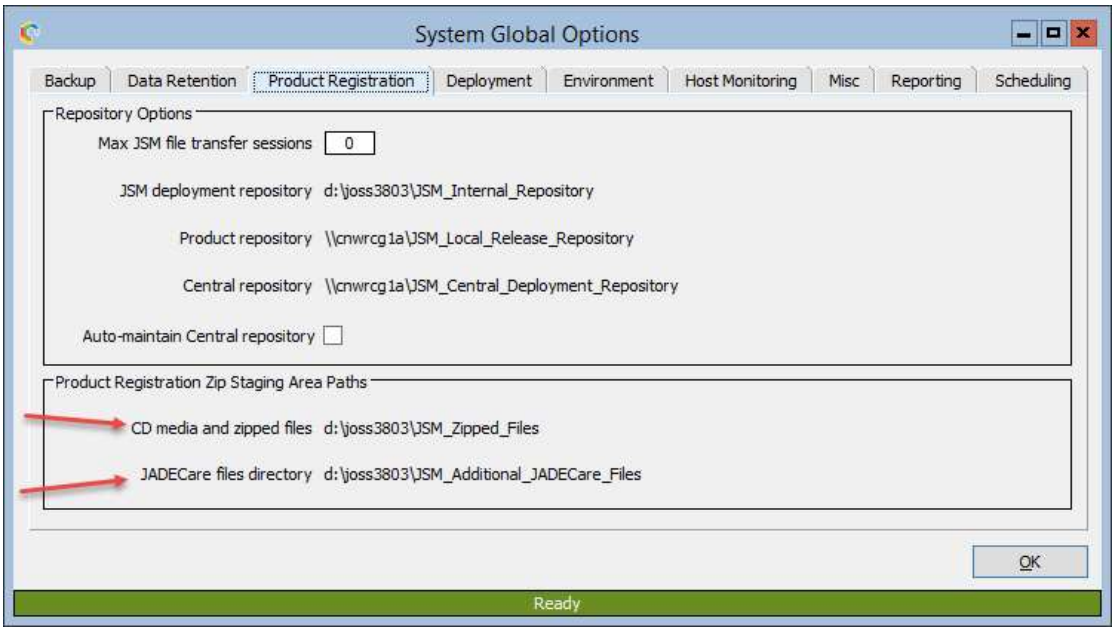
```
JMT
   8.4.02
      i686-msoft-win32
         Base
            a_bin
            a_misc
            a_thinbin
            u_bin
            u_misc
            u_thinbin
         HotFixes
            005
               a_bin
               a_misc
               a_thinbin
               u_bin
               u_misc
               u_thinbin
            006
               a_bin
               a_misc
               a_thinbin
               u_bin
               u_misc
               u_thinbin
      x64-msoft-win64
         Base
            a_bin
            a_misc
            a_schemafiles
            a_thinbin
            u_bin
            u_misc
            u_schemafiles
            u_thinbin
         HotFixes
            005
               a_bin
               a_misc
               a_schemafiles
               a_thinbin
               u_bin
               u_misc
               u_schemafiles
               u_thinbin
            006
               a_bin
               a_misc
               a_schemafiles
               a_thinbin
               u_bin
               u_misc
               u_schemafiles
               u_thinbin
            .
         etc
```

Product Registration

JSM provides the facility to automatically build the product repository. There are two ways to instigate this setup:

- JSM can build the repository automatically
- The repository can be built manually

JSM automatic building of the product repository



CD media and Zipped Files

The directory must be defined in the JSM initialization file **[JadeSystemsManager] ProductRegistrationDirectory** and must be local to the JSM host, once this directory is defined and the 'OK' button is clicked the following directory structures are created under the defined path.

```
CardSchemaReleases
      _SchemaFiles
      i686-msoft-win32
      x64-msoft-win64

HotFixes
      CardSchema
      Jade
      JMT

JadeReleases
      armv4i-msoft-wince50
      armv4i-msoft-wm60
      i686-msoft-win32
      x64-msoft-win64

JMTReleases
      armv4i-msoft-wince50
      armv4i-msoft-wm60
      i686-msoft-win32
      x64-msoft-win64
      JMT_Documentation
      JMT_HTDocs
```

The distribution names are examples of the distribution naming conventions held within the JSM database. The distribution names i.e.: i686-msoft-win32 should contain the entire product release.

Some examples:

- **CardSchemaReleases**

  For a CardSchema release the .scm & .ddb file are placed under the _SchemaFiles folder as these files are common to all distributions. The underscore exists in the directory name so it always resides at the top for the distributions.

  CardSchema binaries are denoted with <name>.ansi or <name>.unicode, both .ansi and .unicode versions of the binaries can reside together in this folder and JSM will take care of the placement when building the physical respository. If the filename contains no .ansi or .unicode extension then the file is copied into both the a_bin and u_bin folders of the physical repository.

- **HotFixes**

  The folder now requires sub-directories, one for each product that can have hot fixes or consolidated releases.

  - CardSchema
    The registration process expects an archive file of the following format and content:
    - c_<hotfix / consolidated release number>_SCH.rar
      the schema and ddb file are expected to be at the root level of the archive file and are the only files that will be copied
    - c_<hotfix / consolidated release number>_BIN.rar
      the files are expected to be within <distribution> directories and, if the file is generic, these will be at the distribution root, otherwise ANSI files are expected to be in a_bin and Unicode in u_bin

  - JADE
    Hotfixes can be downloaded from PARSYS for one or more distributions with the distribution expected to be part of the name, format of name as follows:
    - JADE_<release>_<distribution>-<characterset>_<hotfix version>.zip

      (Example : JADE_7.1.5_I686-MICROSOFT-WIN32-ANSI_7.1.05.003.zip is hotfix 3 for JADE 7.1.05 for Windows 32-bit ANSI environments)

    The registration process will search for and extract all hot-fixes within the JADE hotfix staging area.

  - JMT

- **JadeReleases**

For JADE 7 releases the media format is provided either on DVD or via a network share. The directory format is as follows and the registry process is cognizant of the specific structure of JADE 7 releases. See the section on 'Compact JADE Notes' for details of the directory expectations.

```
JADE 7.x
```

Ansi32
Ansi64
Compact
Unicode32
Unicode64

Compact Jade Notes

Compact Jade releases require some additional steps when placing them in the JadeReleases staging area.

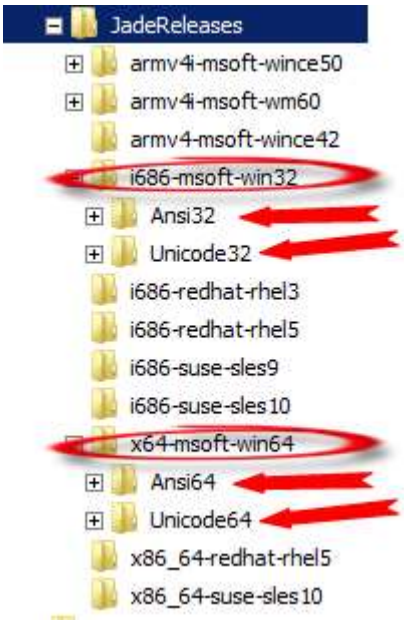
Jade 7 releases

The automatic repository building is slightly different.

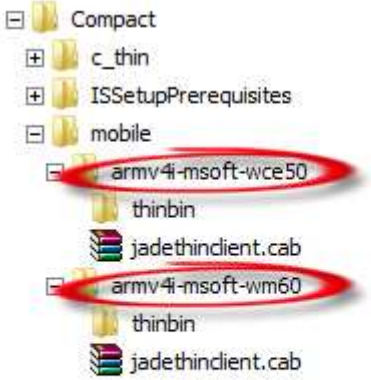
```
JADE 7.x

     Ansi32
     Ansi64
     Compact
     Unicode32
     Unicode64
```

These folders should be copied into the following zipped files location distribution folders



The compact JADE release setup requirements remain the same as previous JADE releases.



the contents of each distribution folder located in the '**Compact\mobile**' directory on the DVD media should be copied to the applicable distribution folder in the zipped staging area.

For the example shown here:

the contents of 'armv4i-msoft-wce50' should be copied to the 'armv4i-msoft-wince50' folder located under the zipped staging area.

the contents of 'armv4i-msoft-wm60' should be copied to the 'armv4i-msoft-wm60' folder located under the zipped staging area.

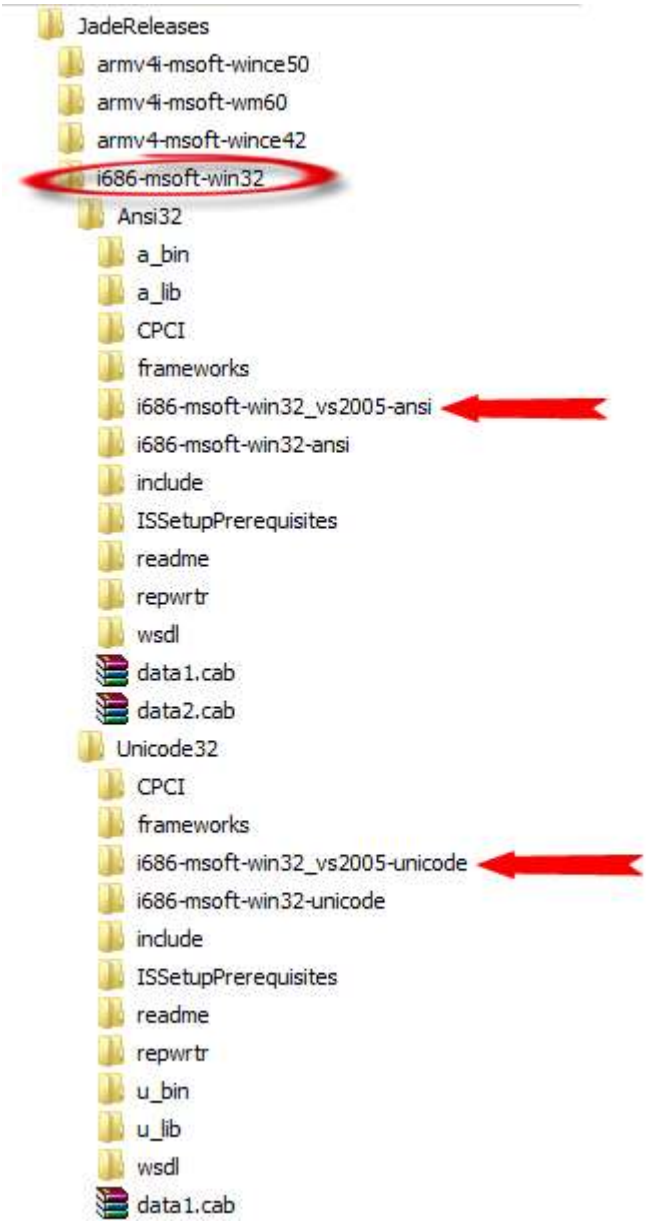
JADE 7.0.05 and later release media for 'Compact' JADE now have a 'download' folder instead of 'thinbin' shown above. The procedure remains unchanged. Copy both the 'download' folder and 'jadethinclient.cab' file into the appropiate distribution directory located in the zipped staging area.

JADE 7.x introduces a new distribution requirement, the new distribution is called:

`i686-msoft-win32_vs2005`

The files required for this distribution are expected under the `i686_msoft_win32\Ansi32` and `i686_msoft_win32\Unicode32` distribution folders.

Providing you have copied down the Ansi32 or Unicode32 folders from the release media there is nothing more to do.



You will not find a `i686_msoft_win32_vs2005` auto-created directory in the zipped staging area root directory as it is not required for the vs2005 distribution.

Additional JADE Care Files

Only applicable to the JADE product releases.

Like the CD Media and Zipped Files structure the Additional JADE Care Files directory contains distribution names:

```
i686-msoft-win32
x64-msoft-win64
```

These are files for a given ProductRelease such as:

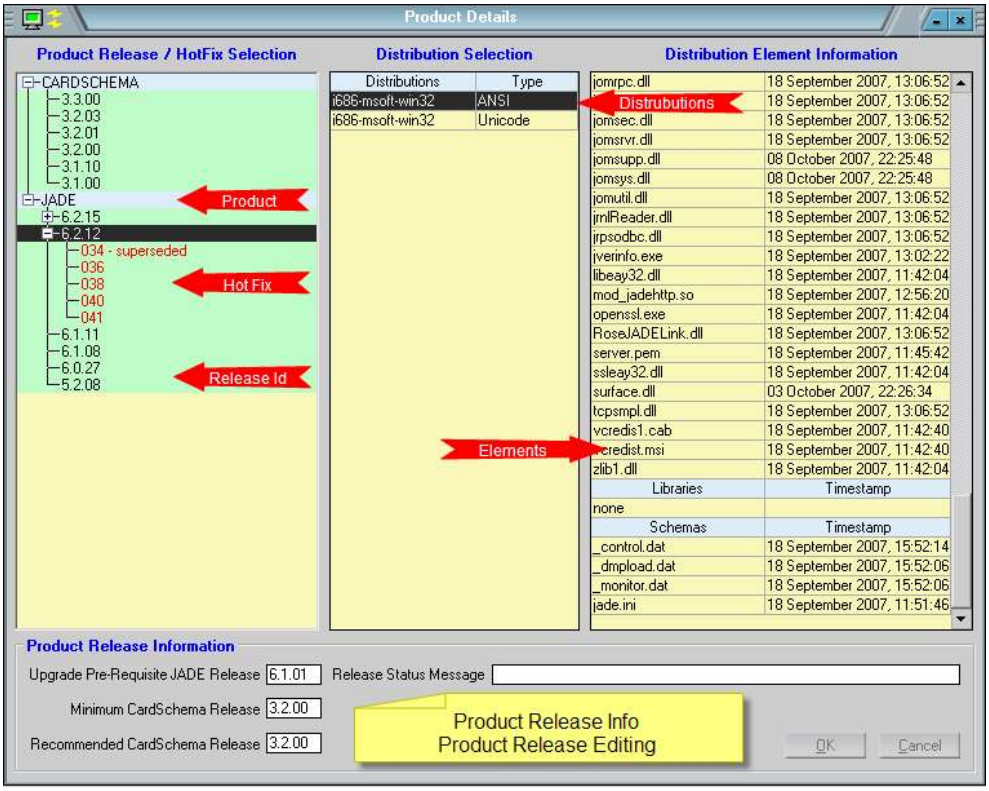
```
cnwplugin.dll
cnwplugin.ddb
jade_preupgrade.bat
jade_preupgrade_files.txt
jade_upgrade.bat
JadeDevSecurity.dll.ansi
JadeDevSecurity.dll.unicode
```

If you've asked JSM to build a physical repository for ProductRelease 6.2.14 then any files located under

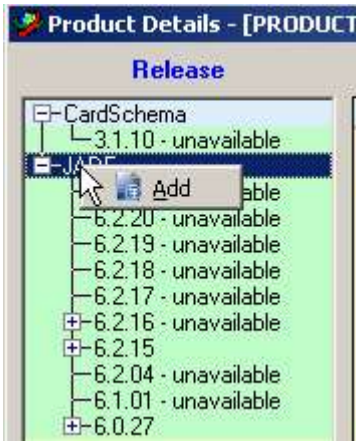
<distribution name>/6.2.14 will also be added into the physical repository. If the file contains a .ansi or .unicode extension it will be located into the appropriate a_bin or u_bin folder. If the file extension is a .scm, .ddb, .cls or .mth then these files will be copied into the _dep_p and _sys_p folders.

Initiating a build of the product repository

Clicking the 'Setup / Products & Product Releases' menu item presents the **ProductViewForm**.



To add a new ProductRelease, select and right mouse click on the desired product i.e.: CARDSCHEMA or JADE



Define your release details and commit the new ProductRelease. Note that the new ProductRelease will be marked as unavailable is it contains no registered product elements or a product element registration has not been completed successfully.

Now select and right mouse click on the new ProductRelease and this popup menu will appear

Select 'Build And Register All' for a complete physical repository build of a new release. JSM will prompt for confirmation then proceed to build the new product repository. JSM will create all the required directory structures and copy any JADECare additional files into place.

The 'Build And Register All' process can be refined further, for example, if you only want to build a repository for Windows 32bit ANSI you can:

1. click the ProductRelease i.e.: 6.3.01
2. select the i686-msoft-win32 ANSI distribution and right mouse click to bring up the popup menu item.

3. click the 'Build And Register All' menu item to build the physical repository just for the selected distribution.

Building the physical repository for a an entire ProductRelease is usually a one-off requirement. Hot fixes however can be frequently released and need to be added to the physical repository. For this reason there is a separate 'Build And Register Hot Fixes Only' menu item.

Any hot fixes located in the HotFixes folder that match the selected ProductRelease will be physically added to the repository. If the hot fix includes a schema then JSM will automatically create the jade_upgrade.bat script for the hot fix also and, in addition, any readme.txt file accompanying the hot fix will be copied to the root folder of the hot fix id. This is to allow operations to check for any special instructions such as the loadStyle of the patch.

Manually adding a new JADE Release to the Repository

To set up the repository for a new, full JADE release, perform the following steps:

1. Create directory **<JADE Release Id>/Base** under the **<Product Repository Top Directory>**.
2. Copy the entire Windows directory from the installation CD to the Base directory, so that you end up with **/<Jade Release Id>/Base/Windows/a_bin** etc.
3. Rename the **a_thin** directory to be **a_thinbin**.
4. If a Linux release:
   - copy directory **<distribution>\download\bin** as **a_thinbin** directory
   - copy directory **<distribution>\download\lib** as **a_thinlib** directory
5. Create new directory **a_sys_p** at the same level as the a_system directory that you have just copied from the CD.
6. Create new directory **a_sys_p** at the same level as the a_system directory that you have just copied from the CD.
7. Copy the entire contents of the **a_system** directory EXCEPT for the **_control.dat** file and **jade.ini** file, into the **a_sys_p** directory.
8. Create new directory **a_dep_p** at the same level as the a_system directory that you have just copied from the CD.
9. Copy the entire contents of the **a_sys_p** directory into the **a_dep_p** directory.
10. The JADECare Systems Management Development Centre will have provided you with **jade_preupgrade.bat , jade_preupgrade_files.txt jade_upgrade.bat** script file tailored for the release that you are loading (if not then contact JADECare support at jsmsupport@jadeworld.com.) Copy this file into the **a_bin** directory.
11. The JADECare Systems Management Development Centre will have provided you with **cnwplugin** library and debug file for the release that you are loading (if not then contact JADECare support at jsmsupport@jadeworld.com.) Copy the library file into the **a_bin** directory and, if provided, copy the debug file into the **a_debug** directory.
12. Log on to JSM as a superuser and ensure that the Product Master Directory has been defined as described earlier in **Defining the Repository Location**.
13. Use menu item **Setup | Products and Product Releases** to access the **Product Details** Form.
14. Select the product **JADE** in the left-hand side of the displayed list box, right mouse click and select **Add**.
15. Enter the JADE release id (e.g. 6.0.18) and complete the other fields then press OK.
14. Select the release you have just added, right mouse click and select menu **Register All**. JSM now parses the directory structure you have just established and populated (including any hotfixes), and updates its database with information about the release content. Wait until this update is complete before proceeding.

    This process also does several checks to ensure that key files that are not released by the JADE Plant but supplied from Systems Development Group are also present. The content of the **a_thinbin** and **a_thinlib** is checked to be

the same as the equivalent code file in **a_bin** or **a_lib** respectively. Note that the **a_thinlib** is only required for Linux releases.

15. To check the contents of the release, select the **Release,** then the **Distribution**, the **Product Release Elements** will be shown in the right-most table.

16. If everything has been registered correctly the **Product Release** can be marked as 'Available' via the popup menu item on the **Product Release**. Once the **Product Release** is available it can then be deployed.

Manually adding a JADE Hot Fix to the Repository

1. If directory **<JADE Release Id>/<distribution>/HotFixes** doesn't exist under **<Product Release Top Directory>** then create it. (eg \JADEReleases\6.3.05\i686-msoft-win32\HotFixes

2. Create new directory **<hot fix number>** (e.g. 007 – preceding zeros are recommended) under the **HotFixes** directory.

3. Create directory **a_bin** and **a_thinbin** under the **<hot fix number>** directory. If the hot fix includes schema files then also create a **SchemaFiles** directory under the **<hot fix number>** directory.

4. Copy all binaries, debug files and any **jadmsgs.eng** file from the exploded hot fix to the **a_bin** directory.

5. If a Linux release, copy all libraries from the exploded hot fix to the **a_lib** directory.

6. Compare the contents of **\base\a_thinbin** and the hotfix **\a_bin** directory and where the files match, copy these from the hotfix **\a_bin** directory to the hotfix **\a_thinbin** directory

7. If a Linux releases then compare the contents of **\base\a_thinlib** and the hotfix **\a_lib** directory and where the files match, copy these from the hotfix **\a_lib** directory to the hotfix **\a_thinlib** directory

8. Obtain the hot fix and explode the archive into a temporary directory.

9. Copy any **.scm** and **.ddb** files, **.cls** or **.mth** files from the exploded hot fix archive to the **SchemaFiles** directory.

10. Copy the generic hot fix **jade_upgrade.bat** script file into the **SchemaFiles** directory, and edit it to reflect the schema file(s) that are to be loaded as part of the hot fix. The generic hot fix **jade_upgrade.bat** file is listed below, with the items required to be edited underscored: Note that special characters are not permitted in the schema file's file name. If you receive a hot fix from JADE with a name such as **HotFix#312.scm** then rename the file to remove the special character.

```
rem  %%%%%    Batch file for JSM load of JADE hot fix   %%%%%%%%
rem  -- This file for JADE 5.2.08 hot fix 312

%ENV_BINPATH%\jadloadb.exe path=%ENV_SYSTEMPATH% ini=%ENV_BINPATH%\%ENV%.ini
schemaFile=%ENV_TEMP%\hotfix312.scm

// Note : the two lines above must appear on a single line
```

11. In the right-hand table of the **Product Details** form, right click on the release to which the hot fix belongs and select menu **Register Hot Fixes Only**. JSM now parses the directory structure you have just established and populated, and updates its database with information about the hot fix content. Wait until this update is complete before proceeding.

12. To check the contents of the release, select the **HotFix** for the appropriate **Release,** then the **Distribution**, the **Product Release Elements** will be shown in the right-most table.

Manually adding a new CardSchema Release to the Repository

CardSchema (JADECare Start) updates are normally only released as full releases i.e. hot fix consolidations and hot fixes are not normally released for this product.

1. Create directory hierarchy **<CardSchema Release Id>/Base/Windows** under the **<Product Repository Top Directory>**.
2. Under the above Windows directory, create directories **a_bin** and **a_system**.
3. From the CardSchema release consistency, copy ***.dll** to the **a_bin** directory. Do **not** rename **karma.dll.ansiXX**.
4. Copy the latest schemacp.exe (available from JADECare) to the **a_bin** directory.
5. From the CardSchema release consistency, copy the **CardSchema.scm** and **.ddb** files to the **a_system** directory.
6. Use menu item **Setup | Products and Releases** to access the **Product Details** Form.
7. If product **CARDSCHEMA** is not listed in the left-hand side of the displayed table, right-click on the table and select **Product | Add**. Enter the product name "CARDSCHEMA" and description "JADECare Start", and press OK.
8. Select the product CARDSCHEMA and right click. Select menu item **Release | Add.** On the displayed Release Maintenance form, enter the CardSchema release id (e.g. 3.0.05) and complete the other fields then press OK.
9. In the right-hand table of the **Product Details** form, right click on the release you have just added, and select menu **Release | Build Element List**. JSM now parses the directory structure you have just established and populated, and updates its database with information about the release content. Wait until this update is complete before proceeding.
10. To check the contents of the release, in the right-hand table of the **Product Details** form, right click on the release you have just added, and select menu **Release | View Element List.**
11. The release can now be deployed.

Setting Up Single User Application Server

JSM supports the use of a singleuser application server and provides all the managed service options available for standard environments

Configuration requirements are as follows:

- Minimum JADE requirement is JADE 2018 plus all hotfixes up to and including 55 (or later)
- Minimum JADECare Start requirement is 4.0.02
- JAWS control file changes (see below)
- Move any server application definitions from ini file section [JadeServer] to [JadeAppServer] section, notably ‘ServerApplication1=CardSchema,KCCMgr’

JAWS control file changes

The following changes are required to the JAWS control file:

```
 [JadeServer]
Launch=false

[JawsOptions]
NoKCDummy=true

[<appserver section>]
Server=singleuser
```

Setting Up For ODBC Environment Management

Introduction

JSM 3.7.01 provided the ability to monitor a MSSQL user-database from a JSA on the same machine, or a remote JSA.

JSM 3.7.03 provided the ability to manage MSSQL databases (system and user-databases) such that these can be backed up and monitored.

The intent is for JSM to be able to manage any database type that implements ODBC. JSM 3.8.06 provides the ability to:

- define backup scripts applicable to (JSM) known database types.
- back up a non-managed ODBC database remotely from a Jade System Agent by executing the backup script applicable to the remote database type.

A **Managed Server** is one that has a Jade Systems Agent (JSA)

A **Managed ODBC Environment** resides on a a managed server and the JSA provides the ability to backup that environment in a similar fashion to a JADE Environment. A managed ODBC Environment can be monitored locally.

A **Monitored ODBC Environment** resides on a server that does not have a JSA. A monitored environment may be:

- monitored remotely from one or more managed servers
- backed up by an appropriate Powershell script from a designated managed server.

'User Database' Backup

Each user-database relates to an ODBC Environment (in JSM terminology) and each has its own backup definition (backup directory, backup schedule, exclusions, inclusions).

'System Database' Backup

In addition to user databases, each MSSQL ODBC Instance contains three **system** databases

- master
- model
- msdb

System databases can be backed according to a single schedule independent of the schedules for the user databases.

These **system** databases are represented by a shadow ODBC Environment named **_system**, one of which automatically exists for each managed MSSQL Instance.

System databases are not displayed in the ODBC Maintenance form. Consequently, the backup definition (backup directory, backup schedule, exclusions, inclusions) for the system database is accessed via the ODBC instance (see the ODBC Maintenance form below for details).

System databases use the same directory structure as that of user databases except that the ODBC Environment name is **_system**.

ODBC Environment Directory Structure

The following directory structure only applies to ODBC Environment's that reside on a managed server.

The root node of the ODBC Instance is derived as follows according to JADECare Standards:

- ODBC Instance name is . (period or default) then the root node is the database type (eg MSSQL, MYSQL etc)
- If the ODBC Instance name is other than the default, then the root node is the ODBC Instance name

Each ODBC Environment is required to have the following directory structure:

```
<volume>:\<ODBC Instance Name>
    |
    -------<ODBC Environment Name>
            |
            -------data             - location of ODBC database files (eg .mdf)
            |
            -------misc             - location of backup.htm file and misc files
                    |
                    -------scripts  - location of scripts used by backup
```

And on the same volume or a different volume is expected the **logs** directory

```
<volume>:\<ODBC Instance Name>
    |
    -------<ODBC Environment Name>
            |
            -------logs             - location of backup logs and other log files
```

SQL Server Security Requirements

JSA may run under any one of the following accounts:

- local JSA account
- domain JSA account
- local SYSTEM account ('NT AUTHORITY\SYSTEM')

The following security attributes are required for whichever account JSA is executing under to facilitate SQL queries and backups:

Security Role	public
User Mapping (need to define for each SQL database being backed up)	db_backupoperator db_datareader public
Securables	Connect SQL – Grant View Any Definition – Grant

NOTE : There is still an issue outstanding with SQL queries against the **system** databases failing to return results using the settings above. This is under investigation.

Setting Up for Amazon Web Services Backups

Introduction

JSM 4.0.2 introduces support for AWS backup and snapshot maintenance functionality.

This feature is not dependent on JSA and relates to JSM only.

Prerequisites

AWS functions may not be available to all users and is dependent on two additional binaries being present in the JSM installation c_bin folder.

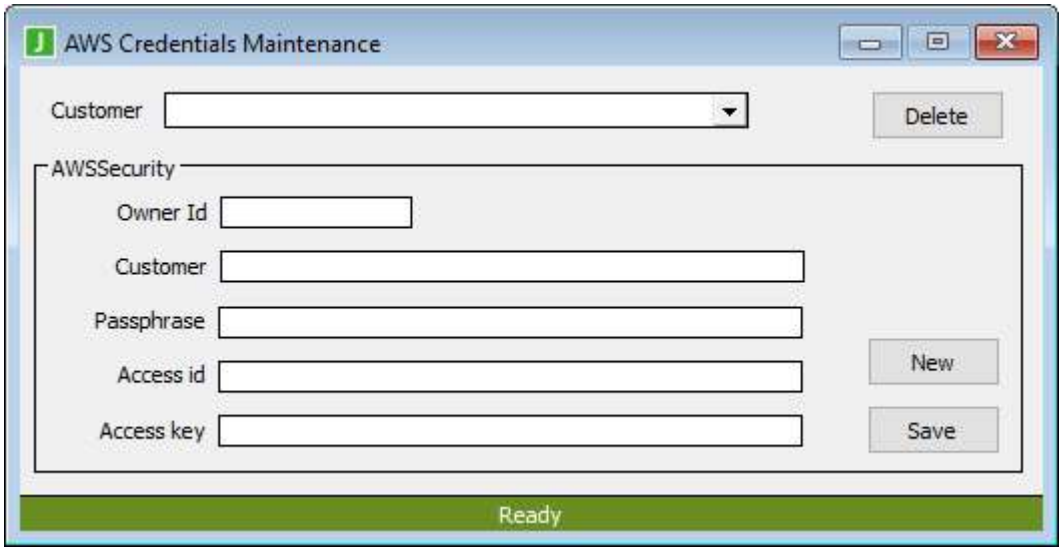
- CnwSkyBound.dll
- CnwSkyBound.pdb

AWS Customer

A unique identifier using AWS provided information for accessing the Amazon Web Services API.

This information can be entered by clicking the 'Setup -→ Amazon Web Services -→ Customer Maint' menu item.

Clicking the 'New' button initialises the form for new customer input.



Customer	Existing customer selection. Note that deleting a customer will also delete everything locally associated with that customer. Nothing will be deleted in AWS.

Owner id	The unique owner id provided by Amazon when you have an AWS account.
Customer	The user provided friendly name for this customer. Any name can be used.
Passphrase	A user provided passphrase to prevent access to the access id and access key information. The intention is that only AWS administrators should know this information and it is up to them to set a passphrase to prevent editing and viewing of it. It is also important to note that the passphrase can not be decrypted (one-way encryption). The passphrase then encrypts the access id and access key. Any exceptions or stackdumps that reference the access id and access key properties will still be encrypted in the output.
Access id	Derived from setting up an AWS account.
Access key	Derived from setting up an AWS account.

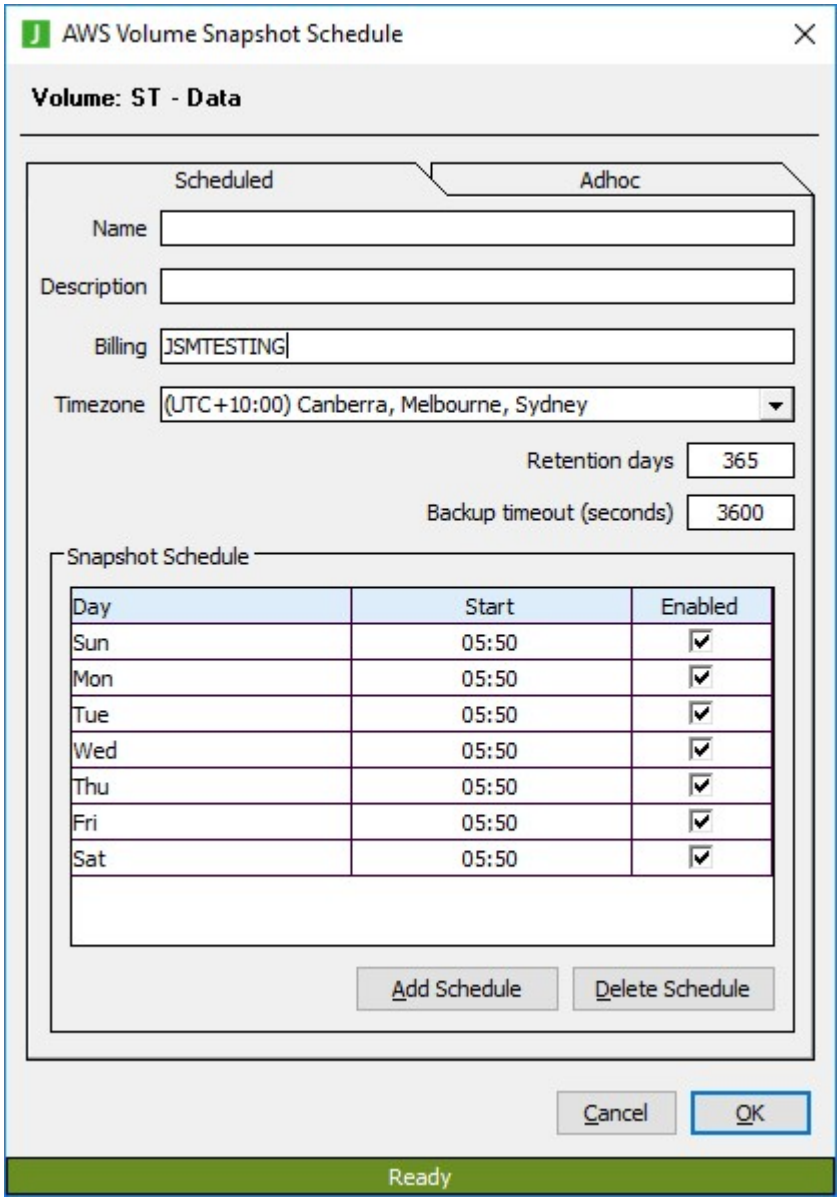
AWS Snapshot

A snapshot (backup) of a volume running on a virtualized host running in the Amazon Web Services environment.

- Scheduled backups
- Adhoc backups
- Snapshot tagging
- Snapshot housekeeping
- Synchronized inventory with AWS as the master.

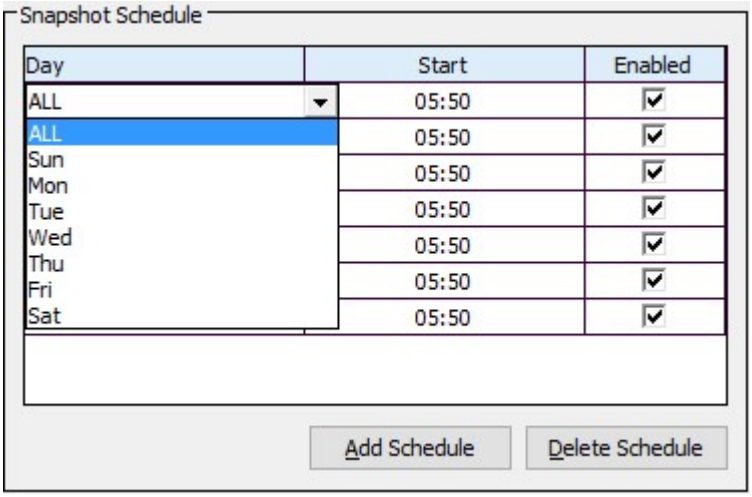
AWS Snapshot Scheduling

Scheduling can be set by either selecting ‘Modify Schedule’ from the popup menu on an existing item in the ‘AWS Snapshot Schedule’ form or by right clicking on an instance in the ‘AWS Snapshot Summary’ form.



Schedule times should be entered local to the selected timezone. In the example above, snapshots will occur everyday at 5:50am in the selected Australian timezone but JSM will start the snapshots each day at 7:50am NZST.

When creating a schedule you can quickly enter a weeks worth of entries or update the whole schedule by selecting ALL from the top row



then change the time next to that cell and finally ticking or unticking then re-ticking the 'Enabled' textbox in the last cell. This will then update the entire schedule with the starttime specified in the first row.

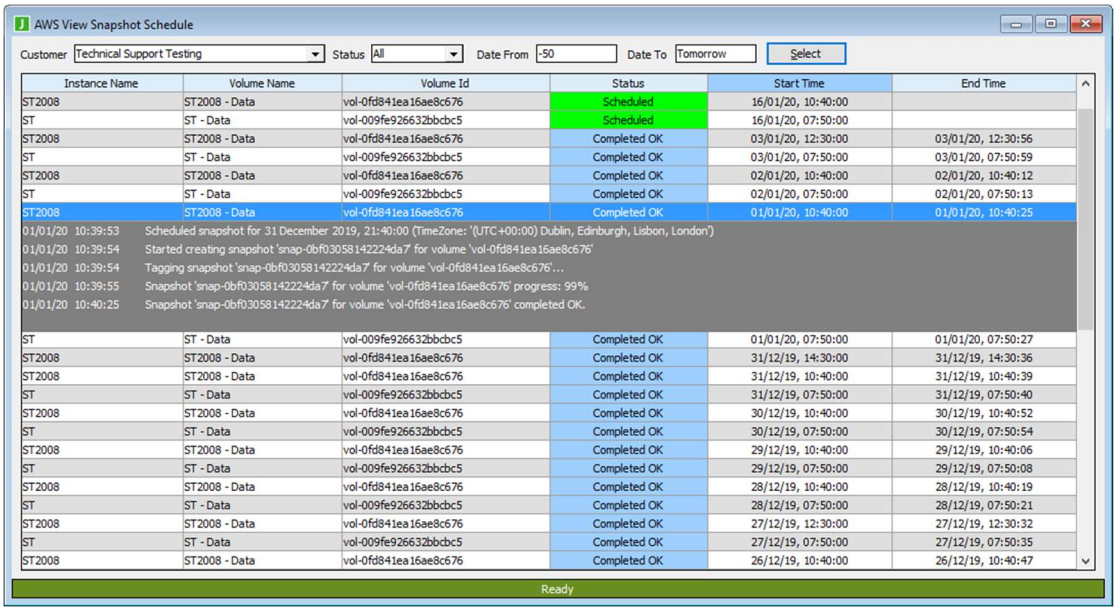
Snapshots can be scheduled for multiple times per day so you can continue to add days and times as you require.

The snapshot schedule form can also be used to view adhoc one-off scheduled snapshots, this type of backup can be canceled from this form but no editing of the schedule is possible.

Changes to the schedule aren't committed until the OK button is clicked.

AWS Snapshot Schedule Overview

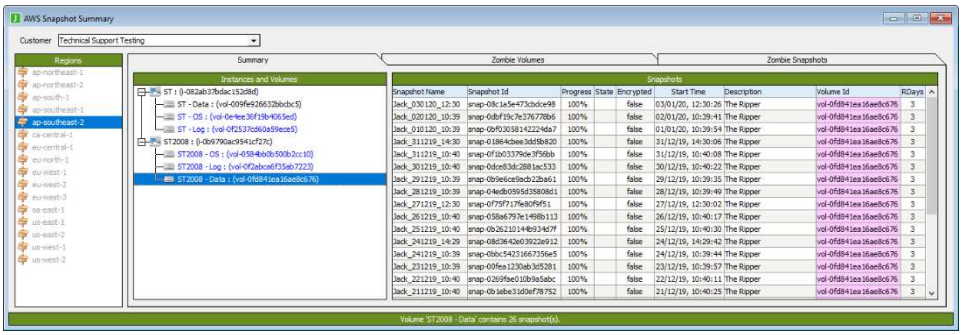
An overview of all snapshot operations for a given customer can be viewed by clicking the 'Setup -→ Amazon Web Services -→ Snapshot Schedule' menu item.



From this form the popup menu items can be used to modify the schedule, cancel a scheduled snapshot and run an immediate adhoc snapshot of the selected item using the existing parameters of the selected item.

AWS Snapshot Summary

From this form you can manage everything to do with your AWS snapshot inventory. This form is access by clicking the 'Setup -→ Amazon Web Services -→ Snapshot Summary' menu item.



The main purpose of this form is to manage everything to do with snapshots and to bring all the disparate components of regions, instances, volumes and snapshots into one easy to view interface.

Hovering the mouse over most items will present further information via bubble-help.

Regions

These are provided by AWS and are not user definable. When JossAWSManager process starts a complete inventory will be retrieved from AWS for each defined account. Regions will be populated during this process.

Instances and Volumes

In most cases volumes are attached to an instance. Instances and Volumes cannot be created/managed or deleted through JSM and must be defined through the Amazon Web Services admin interface.

Snapshots

Snapshots can be created, scheduled and deleted. In addition to this, existing snapshots can have some attributes changed, such as the number of retention days.

These functions are all accessible by popup menu.

Zombie Volumes

Volumes not attached to an instance.

Zombie Snapshots

Snapshots not attached to a volume.

Systems Management Activities

Deletion of JSM Entities

The deletion process causes all references to these items to be removed from JSM visibility and, if not already active, spawns a background application **JanitorDeletionMgr** that deletes the JSM entities.

Jade Environment

A database or remote Jade environment can be deleted by the following means:

- the **Jade Environment Maintenance** form by use of the Delete button.
- Deletion of the master environment
- Deletion of the server host the Jade environment is running on.

Where a database Jade environment is being deleted, all remote Jade environments associated with the database Jade environment are also deleted.

If the database Jade environment being deleted is the only Jade environment defined for a master environment, then the user is asked if the master environment entity should also be deleted.

Master Environment

A master environment may be deleted by:

- Via the **Master Environment Maintenance** form using the Delete button.
- Deletion of the server host the Jade environment is running on when there is only one Jade environment declared for the master environment and the user OK's the deletion of the master environment.

Deleting a master environment deletes all related database Jade environments and associated remote environments as well as JSM maintained directories used to hold the version control files and the environments deployment staging area.

Server Host

A server host may be deleted via the **Host Maintenance** form by using the Delete button.

This causes any database and remote managed environments located on the server host to be deleted. When the deletion of a database managed environment is the last managed environment defined for a master environment, the user is asked if the master environment concerned should also be deleted.

If the server host is a node in a cluster host, then the user is advised and asked whether or not the cluster host should be deleted.

Cluster Host

A cluster host may be deleted via menu item on **System View** form when cluster host is selected.

This causes all nodes of the cluster to be deleted and any database and remote managed environments located on the server host nodes to be deleted. When the deletion of a database managed environment is the last managed environment

defined for a master environment, the user is asked if the master environment concerned should also be deleted.

Licence Key Validation

Any licence key details entered by JSM functions (currently JARI and Restore functions) will be validated against a local registration database maintained expressly for that purpose.

There will be as many registration databases maintained as there are supported major JADE releases, these being retained under sub-directories <JSM root>\server\reg<nn> where <nn> signifies the major JADE release (eg <JSM root>\server\reg160 will be used to validate licence keys for a JADE 2016 environment).

The registration databases are built by the JADE release registration process.

Managed Backups

There are three types of backup:

- Online – true online backup which allows transactions during the backup
- Offline – database is run in singleuser mode and is quiesced
- Journaled – an existing database image is required and journals are replayed updating the database and leaving it in recovery mode.

Online and offlline JADE backups canm be specified with verify and / or compression being selected.

ODBC environment backups are always online and the verify, compress options are not available.

The backups can be scheduled by managed environment for specific days of the week and any times during those days.

Managed backups are controlled by the KarmaBackupMgr application that executes in the managing JSA. Backup of the managed environment's database is performed by the KCOnLineBackup application within the managed environment itself (for this reason, an offline backup is performed as an online, quiesced backup).

Journaled Backups

Journaled backups have been introduced to take advantage of cloud-based backup strategies and to reduce the amount of change between backups allowing dedup processes to function to best advantage.

Journaled backups is an option that can be specified by JADE Environment. The option is only available if the following are all true:

- Target JADE Environment has requisite JADE release installed:
  - JADE 2016 SP1 and hotfix 041 installed
  - JADE 2018 and hotfix 080 installed
  - JADE 2020 and hotfix 011 installed
  - Release later than JADE 2020 installed
- Target Environment has requisite CardSchema release installed:
  - 3.8.10.008
  - 3.9.03.015
  - 4.0.02.009
  - 4.0.03
  - Relase later than 4.0.03
- JSA version is 4.0.03 or later

For a journaled backup to be performed the following pre-requisites must exist:

- The requested backup must have specified online (if scheduled backup) or journaled if adhoc backup request.
- An online backup image must exist in the standard backup staging area
- All journals from the last journal defined in _backup_OK.htm file through to the current abort journal must be present on disk.

If the above pre-requisites are all true, then a journaled backup will be performed, details below.

If the request arose from an adhoc backup, then th backup will fail with the failed pre-requisite stated in the error message.

If the request arose from a scheduled online backup, then an alert will be generated advising that a pre-requisite was missing and the backup will continue as an online backup.

A journaled backup performs the following functions:

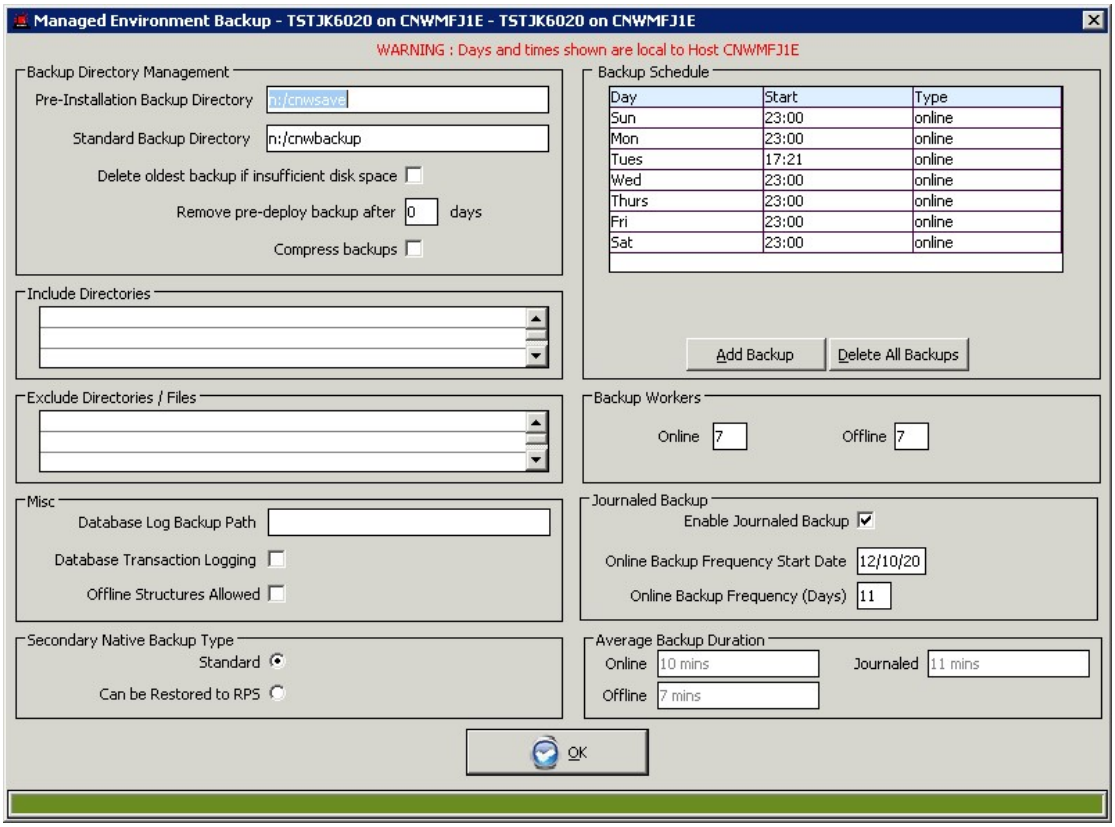
- Copies the requisite journals into the backup staging are.
- Performs a database recovery to all but the latest journal and leaves the database in recovery mode.
- Copies changes to non-database files into backup area.
- Deletes files and/or directories from backup image that are no longer in running environment
- Performs a 'convertToBackup' to:
    - Perform an integrity check on the backed up database
    - Create a new 'backupinfo' file for use in any potential restore.
- Creates a new _backup_OK.htm file

Defining Managed Backup Parameters

The following form is available via:

- the Jade Environment Maintenance form by clicking the Schedule button provided on the Backups sheet.
- The ODBC Maintenance form by selecting the Instance or Environment and selecting the menu item 'Edit ODBC Environment <name> backup specs'
- The System View form by selecting the Environment and selecting the menu item 'ODBC Environment Backup Specs'

The above form is for a JADE Environment.



The above form is for an ODBC Environment. As various elements are not available for ODBC backups, these have been disabled or rendered not visible.

Pre-Installation Backup Directory

This defines the directory location of the backup staging area used for pre-deployment backups and is required to be defined before a deployment can be started.

For Windows O/S the drive letter is required. If the directory provided is **e:/cnwsave** and the environment being backed up is CNWTEST then the backup directory root for this environment will actually be **e:/cnwsave/cnwtest**.

For Linux O/S the directory is a single node. If the directory provided is **cnwsave** and the environment being backed up is CNWTEST then the backup directory root for this environment will actually be **/opt/jsm_backup/cnwtest/cnwsave**.

Standard Backup Directory

This defines the directory location of the backup staging area used for managed backups and post-deployment backups.

For Windows O/S the drive letter is required. If the directory provided is **e:/cnwbackup** and the environment being backed up is CNWTEST then the backup directory root for this environment will actually be **e:/cnwbackup/cnwtest**.

For Linux O/S the directory is a single node. If the directory provided is **cnwbackup** and the environment being backed up is CNWTEST then the backup directory root for this environment will actually be **/opt/jsm_backup/cnwtest/cnwbackup**.

For ODBC environments on a managed server, the backup root directory is derived from the backup path name specified and from the names of the instance

and environment. If the backup path is **t:\cnwbackup** then the following derivations can be made:

- If instance is default name (.), the environment name is TSTJK602R_RPS and database type is MSSQL, then derived backup path is **t:cnwbackup\MSSQL\TSTJK602R_RPS**
- If instance is testing, the environment name is SQL_102 and database type is MSSQL, then derived backup path is **t:cnwbackup\testing\SQL_102.**

Delete Oldest Backup

This option is not applicable for ODBC Environments.

When a managed backup or deployment has insufficient space for a backup, setting this option allows the oldest backup on disk to be deleted, these being held in the 'pre-installation backup directory' and 'standard backup directory' and to take this into account when doing disk free-space checks prior to starting the backup or deployment.

When insufficient space exists for a backup and this option is set, then if backups exist in the pre-installation directory and the standard backup directory and these reside on the same disk volume then the oldest of these will be removed. Note that the disk space check will be repeated again after such deletion to ensure adequate disk space exists for the backup.

Remove Deploy Backup after <n> days

This setting enables specification of how long JADE pre-deploy backups are kept before they are deleted by the daily housekeeping application.

Compress Backups

This flag identifies whether a JADE Environment backup is to be compressed or not.

This option is not available for ODBC environments as the ODBC Script used defines whether the backup is compressed or not, by default all ODBC backups are compressed.

Include Directories

For a description of inclusions, see subsection "Include External Directories"

Exclude Directories / Files

For a description of exclusions, see subsection "Exclusion of Directories, Files, File Types".

NOTE : Backups initiated by the deployment process can have additional exclusions defined by the use of ini file settings under section [JSMDatabaseBackup] with verb DeployBackupExclusion<n> where n can be from 1 to 99.

Offline Structures Allowed

This option is not applicable for ODBC Environments.

This setting will suppress the alerts that can be generated when a database structure is detected as offline.

This is to support the **CNWREFRESH** process which allows an environment to be replicated from another and may refresh only some of the partitions, keeping the 'refresh' package and target environment smaller than its originator.

Secondary Native Backup Type

JSM provides the ability to take a backup of an SDS Secondary Native database suitable for restore to an RPS node (by utilizing this ability provided in JADE 2018). This was implemented in JSM 4.0.01.

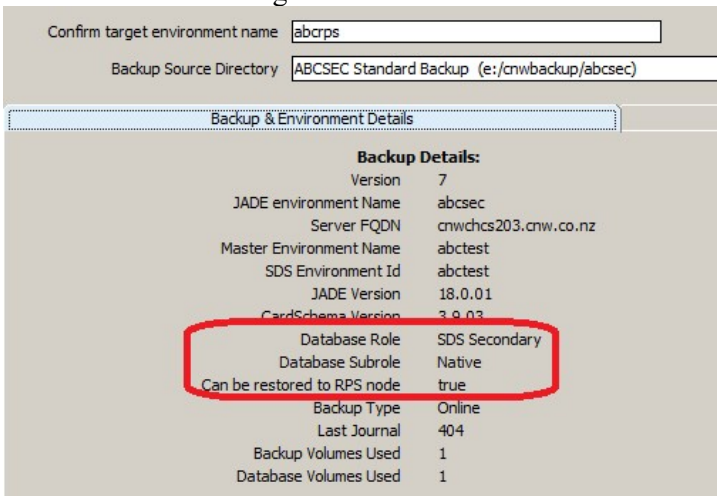
A requirement for this purpose is that the backup captures a quiet point on the Primary database after backing up the Secondary database files. Because the request for a quiet point can time out, ini file settings are provided that enable the backup commit (and quiet point request) to be retried. Refer to:

[JSMDatabaseBackup]
CommitCoherentBackupMaxTries
CommitCoherentBackupRetryInterval

…earlier in this document.

If all attempts fail then the backup will complete with a standard commit, meaning that the backup is suitable for restore to itself, but not an RPS node. In addition, either a tracker call is opened, or an alert is created.

Tip: When the 'Can be Restored to RPS' option is specified, the suitability for RPS restore is recorded to the backup history log viewable in either the View Backup Schedule or Activity Log forms.
In addition, when performing a restore, the backup information displays suitability for restore to RPS along with the database role and subrole of the backup image:



Tip: to guarantee that an SDS Secondary Native backup is suitable for RPS restore, perform an ad-hoc backup and specify 'Restore to RPS Required'. When this option is specified, the backup commit will retry until a quiet point is achieved, or until the backup is aborted.

Note: Pre-deploy backups do not use this feature and are not suitable for restore to an RPS node in the SDE.

Backup Workers

This option is not applicable for ODBC Environments.

CardSchema supports multiple asynchronous backup workers running for a database backup.

The number of backup workers can be varied for online and / or offline backups (the default value for backup workers is 1). The number of workers affects how many concurrent database mapfiles can be concurrently backed up into the backup staging area.

Note that the number of workers specified here takes effect for scheduled, ad-hoc and pre- and post-deploy backups. The only backup scenario that is not handled in this manner is following a deploy where no-reorg was required and no JADE version upgrade occurred.

Note that the offline backup workers specified is what is used by the deployment restore process and is provided as the default value for a user initiated restore of a host environment.

Journaled Backup

This group box and all controls within will be disabled until such time as all the pre-requisites exist that allow 'journaled backup's to be performed.

These are:

- Target JADE Environment has requisite JADE release installed:
  - JADE 2016 SP1 and hotfix <xxx> installed
  - JADE 2018 and hotfix <xxx> installed
  - JADE 2020 and hotfix <xxx> installed
  - Release later than JADE 2020 installed
- Target Environment has requisite CardSchema release installed:
  - 3.8.10.008
  - 3.9.03.015
  - 4.0.02.009
  - 4.0.03
  - Release later than 4.0.03
- JSA version is 4.0.03 or later

Enable Journaled Backup

If checked then this enables journaled backup methodology effective from the Start Date, see next.

Those scheduled backups specified as online will default to being 'journaled backup' if the pre-requisites exist (being a valid online backup is in the standard backup area and the number of days since last online backup is not exeeceded).

Online Backup Frequency Start Date

The online backup frequency defined is only to apply from this start date.

Online Backup Frequency (Days)

This specifies the number of days that may pass before an online backup must be performed. This is required as there is no other way, currently, of ensuring the integrity of the online database.

Average Backup Duration.

There are 3 sets of durations matained by the backup engine. Each type of backup will retain the last 5 durations. These are averaged and form the basis for estimating how lomg a backup is expected to take.

For information only.

View Progress

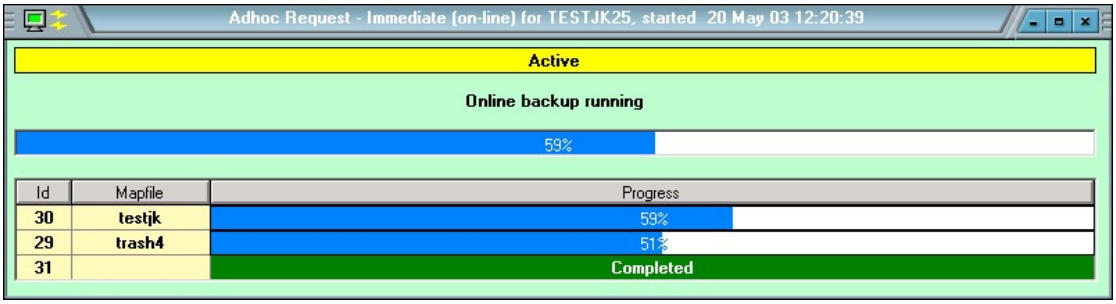
This form can be viewed via the Schedule Backups view or from the Deploy Control form.

The Schedule Backups form requires that you select the row pertaining to the backup you wish to see the progress of, right click to obtain a menu and select the 'View Progress' menu option.

In the Deploy Control form, the progress bar will be shown when one of the backup phases is active (yellow) by double clicking in the column concerned.

The following form is a snapshot of the progress form during the database backup phase where the number of online workers specified was 3. The primary progress bar indicates how far through the database backup is and the progress bars shown in the table provide an indication of how far the backup workers are through the map file they are currently backing up.

In the example below, worker id 31 has completed as there are no more map files to backup whereas workers 29 and 30 are still in the process of backing up map files trash4 and testjk respectively.



Backup Schedule

The Backup Schedule form can be obtained from the following sources:

- Menu item View | Backup Schedule
- System View form by clicking on Jade Environment or ODBC Environment and selecting menu item 'View Backup Schedule'
- From Jade Environment Maintenance form using button 'View Sched' on Activities / Backups sheet.
- From ODBC Maintenance form on selected instance or environment and selecting menu item 'View Backup schedule <entity>'.

The following form can be used to view the backup history for a variety of criteria:

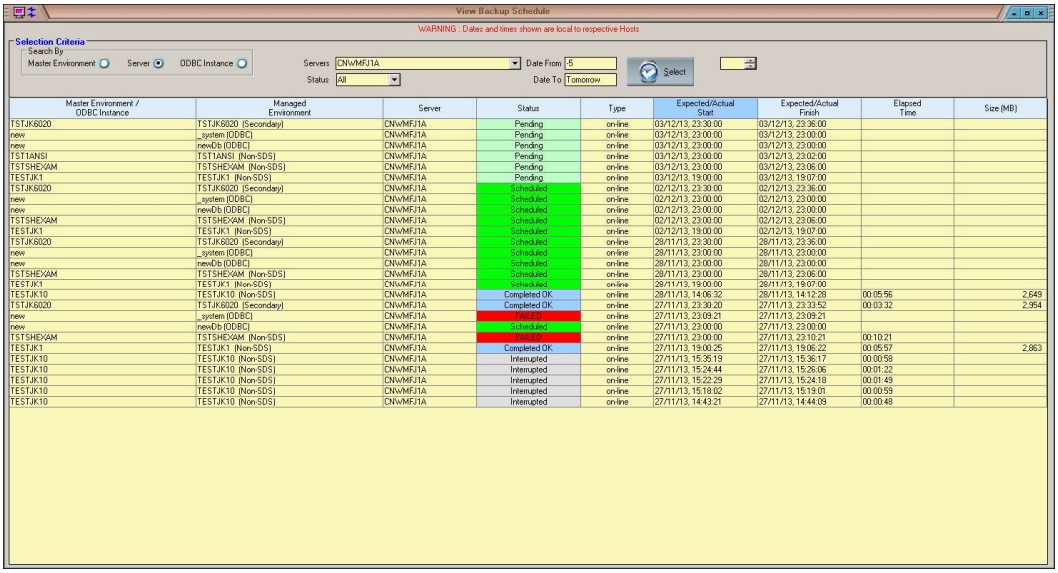
- By JADE Master Environments – all or specific
- By server - All managed environments for all servers or for a specific server
- By instance – All ODBC environments for all instances or for a specific instance

The criteria can be further narrowed down as follows:

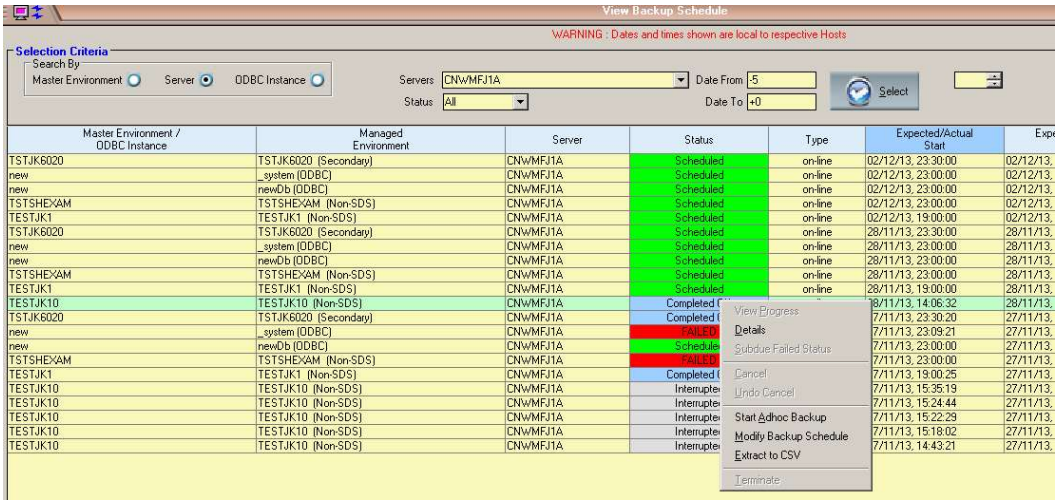
- A specific status or all statii
- Date Range

The date range can be:

- Date From
  - A valid date
  - The literal Yesterday
  - A hyphen followed by a number indicating how many days ago from today (-5 equals todays date minus five)
- Date To
  - A valid date
  - The literal Tomorrow
  - A plus sign followed by a number indicating how many days ago from today (+2 equals todays date plus two)



By right clicking in the table, a list of menu options is provided. If a row has been selected (left-click and shows highlighted by a light-green background), many of the menu options are enabled.



Extract to CSV

The contents of the backup schedule shown will be extracted to a comma-separated-values file on the presentation client to directory c:/temp, the file name being displayed on completion of the extract.

View Progress

See paragraph below which discusses the View Progress option (which is also available from the Deploy Control Form)

Details

This opens an extra row containing a mini-log of information about the backup. Double-clicking on the row provides the same facility.

For detailed information about the backup, these details are contained in the b_log files created for each environment backup and resident under the environment's log directory.

Subdue Failed Status

This allows an operator to change the red failed status to an orange state to indicate that the backup failure has been reviewed. The details of who subdued the failed state is logged and visible in the mini-log (see Details above).

Cancel

Clicking on Cancel causes the selected scheduled or pending backup to be cancelled. This backup can be reinstated by using the Undo Cancel option described next.

Undo Cancel

Selecting this option for a cancelled backup, causes the specified backup to be re-scheduled and undoes the Cancel requested earlier.

Start Adhoc Backup

This allows the operator to start or schedule an adhoc backup for the host environment in the selected row.

Terminate

Allows the operator to interrupt a running backup. Details of this request are logged and the final status of such a backup is Interrupted.

Backup Uses Mount Points

This feature is first available in the 3.9.03 release. It has only been implemented for JADE Environment Managed Backups.

The feature is designed to allow one or more JADE Environment backups to define the backup directory structure where either the backup root directory or one or more sub-directories can be defined as a mount point to another volume. This uses the MS-DOS command **mountVol**.

This feature is implemented when the ini file of the environment to be backed up contains **[JSMDatabaseBackup] BackupUsesMountPoints=true**.

When this ini file option is set then:

- A script titled **buildBackupDirectoryStructure.cmd** or **buildBackupDirectoryStructure.ps1** must be resident in directory <env root>\server\c_misc\Cardinal.
- The backup and deployment engines will NOT perform any free space checks on start up.

It is expected that the execution of the script will, at minimum, create the <backup root> directory and ensure appropriate permissions are defined, if required. See script examples below and notes associated with each.

If the script is not resident the backup will fail.

If the <backup root> directory has not been created then the backup will fail.

Example #1

The following script is a CMD type file and will define the entire backup root directory as a mount point. It specifically:

- defines the backup root directory
- sets the backup root directory to a mount point (in this case volume J: or K: depending on whether predeploy backup or not)
- creates a sub-directory of <backup root>\server\c_system.

The script is soft-coded to use set variables inserted by the JSA backup engine to cater for any JADE Environment and caters for pre-deploy backups and all other types.

In the case below, the JSA backup engine will define the appropriate permissions.

```
md "%env_backuppath%"
if /I %env_backuptype% EQU predeploy goto :PREDEPLOY
rem mount interim backup directory to volume j:\
mountvol "%env_backuppath%" \\?\Volume{71083326-e067-11e8-9d40-
002186edcd01}\
goto :DEFINE_SUBDIR

:PREDEPLOY
rem mount interim backup directory to volume k:\
mountvol "%env_backuppath%" \\?\Volume{05d5c844-00b3-11e4-9af6-
806e6f6e6963}\

:DEFINE_SUBDIR
md "%env_backuppath%\server\c_system"
```

Example #2

The following script is a CMD type file and will define the backup root directory with only the <backup root>\server\c_system being defined to use a mount point. It specifically:

- creates a sub-directory of <backup root>\server\c_system.
- sets the <backup root>\server\c_system directory to a mount point (in this case volume J: or K: depending on whether predeploy backup or not)
- specifically defines the permissions for the mount point volume to ensure that the JADE Environment backup processes can write to this volume.

The script is soft-coded to use set variables inserted by the JSA backup engine to cater for any JADE Environment and caters for pre-deploy backups and all other types.

In the case below, the JSA backup engine will define the appropriate permissions for the <backup root> directory and sub-directories but the mount point volume will need to have explicit permissions defined otherwise the database backup processes initiated by the target environment will be unable to access the destination path and fail.

```
set volume_predeploy=s:\
set volumename_predeploy=\\?\Volume{2837524f-ca05-11e0-b793-
00219b5db84e}\

set volume_standard=r:\
set volumename_standard=\\?\Volume{bd5aa080-eb61-11e8-9634-
00219b5da91b}\

md "%env_backuppath%\server\c_system"

if /I %env_backuptype% EQU predeploy goto :PREDEPLOY

rem mount interim standard backup directory to volume R:
set current_volume=%volume_standard%
set current_volumename=%volumename_standard%
goto :DO_WORK

:PREDEPLOY
rem mount interim predeploy backup directory to volume S:
set current_volume=%volume_predeploy%
set current_volumename=%volumename_predeploy%

:DO_WORK
del /F /S /Q %current_volume%
mountvol "%env_backuppath%\server\c_system" %current_volumename%
icacls %current_volume% /t /q /grant SYSTEM:(OI)(CI)F
Administrators:(OI)(CI)F %env%:(OI)(CI)F %env%_g:(OI)(CI)RX
%env%_m:(OI)(CI)RX
```

ODBC Backup Script Management

Implementation of the ODBC backup feature requires the existence of a PowerShell backup script that when executed performs the backup. In general, the JSM script management feature enables a user to define a script by name and load the script source from file.

In the case of ODBC backup scripts, an associated script exists for each JSM-known database type having the same name as the database type. Note : the default script will execute correctly but always return an error causing the backup to be marked as failed. The default script should be overwritten with a valid script before any backups are implemented.

ODBC backup scripts are located under script activity ODBC Backup in the Script Maintenance form. Backup scripts are automatically distributed to Jade System Agents that perform backups.

As part of the migration to 3.7.03, JSM is preloaded with a script object for ODBC database type MSSQL and are stored against preloaded activity “ODBC Backup”. The preloaded script does **not** have any script source assigned. The script source is loaded using the Script Maintenance form (refer to Script Maintenance section later)

ODBC Backup Script Requirements

PowerShell backup scripts responsible for backing up any ODBC database are expected to issue standard output as the backup progresses. The managing Jade System Agent interprets, logs and processes this standard output.

A line of standard output is interpreted as a tuple of space separated words. The first word in a line is a key value that identifies the context of the output line. Subsequent words are interpreted as values in relation to that context.

The ODBC backup script must produce a line starting with a known key value in order for the Jade Systems Agent to interpret it as backup advice (otherwise the line will be logged with no further action being taken).

Current key words and example value words are:

Word 1 (key word)	Word 2 (value)	Word 3 (value)	Word 4 (value)	Usage
"BackupPath"	<Backup Path>			Logging only
"DatabaseKey"	<Host name>	<instance name>	<server name>	Logging only
"BackupComplete"	NA			Logging only
"BackupStarted"				Feedback to JSM
"PercentProgressUserData"	<percent done>			Feedback to JSM
“PercentProgressSystemFile”	<System file name>	<percent done>		Feedback to JSM
“SystemFileBackupComplete”	<System file name>			Feedback to JSM

Recognized ODBC Backup Script Error Codes

The user defined ODBC database backup script is executed by karma.dll on behalf of the Jade Systems Agent managing the backup schedule. The backup script source may define return codes (e.g. by Powershell syntax “exit <error code>”) that karma.dll passes back to the Jade Systems Agent when the script exits. The following error codes are paired with the given error text that is then logged and passed back to JSM.

Return error code	JSA error text
1	Database defined as Mirror
2	Backup Path not defined
3	Database Backup failed
4	Log Backup failed
5	Unable to find credentials
6	Decryption Error
7	SQL Error
8	Backup Cancelled

Note that recognized error conditions that prevent karma.dll from executing the script will be logged as a negative integer in the range -1 to -16. Consult karma.dll documentation for the meaning of each of these error codes.

Multiple Backup Volumes

It is now possible to specify individual database structures be backed up to separate backup volumes other than the default backup volume. This is done via the Host Environment Maintenance form on the sheet Database Structures. This is available on Windows operating systems only.

Compaction

This facility is no longer available as part of the backup as compaction may now be done online.

Retention of backup directory

In earlier releases, the backup directory was deleted (which may have contained a complete and recent copy of the environment) when the backup process started. This meant that if the current backup failed and issues were found with the 'live' environment, then the only recourse was to restore the environment from a tape copy.

As from the 2.9 release, the backup directories as defined in the Host Environment Backup form are retained until a successful backup has been taken. The backup process (regardless of scheduled, ad-hoc or deployment) will now be taken into an interim directory (<backup directory>_temp/<env>) and when successfully completed, will remove the previous backup directory and rename the interim directory to the final backup directory name.

Note that permissions and shares (including cluster shares) are retained.

WARNING : *Cluster File Shares defined will now be marked as offline and online during the last phase of a backup. Care must be taken to ensure that other cluster resources are not dependent on a File Share resource.*

Dependent on the management software installed on the machine (Compaq Cluster monitoring for example) the act of taking the File Share offline and bringing it Online may generate SNMP traps indicating that the state of a ClusterGroup or Resource has changed or become degraded etc. So this will need to be documented clearly so Operations staff do not attempt to bring the share back online whilst JSM requires it to be offline.

Compressed Directories

As from the JSM/JSA 2.9 release, the backup process detects if the directory nominated as the backup directory is compressed. It ensures that the interim backup directory (<backup directory>_temp/<env>) is compressed if the final destination is defined as compressed.

If the final destination directory (if the final destination directory is not present, then the parent root directory is checked) is compressed, then the free space check algorithms are modified to use the compression ratio from the last successful backup that was written to a compressed directory.

Transaction Journals

As these are copied, the source file is used to create an MD5 hash value that is compared against that derived from the copied file. This is to ensure the media integrity of the transaction journal.

WARNING:

Up until JADE 6.1, the backup provided by the JSM Managed Backup Service has been a complete and independently usable backup which can by used to restore to a working database.

For SDS Secondary databases using SyncMode=JournalBlockWrite (available in JADE version 6.1) an online backup will backup all but the last journal in-use at the time of the backup. This is not an issue if the backup is to be used to restore the SDS secondary.

It **IS** an issue if the backup is to be used to propagate a non-SDS database or an SDS primary. This can be detected by the presence of sentinel file **_backup_requires_journal.htm**, which contains the journal number required.

Include External Directories

External directories (directories other than the JADE system environment) may now be specified and will be added to the files backed up.

These external directories must exist and be available for backup at the start of any backup and deployment.

Each inclusion for an environment running on a Windows platform must declare the drive letter and the environment name as the minimum specification. (eg f:\sampleenv and d:\sampleenv\images indicate different directory levels to be included in the backup). Note that cluster and file shares for included directories will be closed at the <drive>:\<environment> level for each inclusion regardless of the number of nodes specified in the inclusion.

The files from each included directory are copied after the database, transaction journals and JADE system's binaries and miscellaneous files have been successfully copied. Any file shares open to these directories are closed immediately before each directory is copied.

Where an include directory is part of a backup, the backup directory will be formatted as follows:

Windows:

<backup area>/<env id>/_drive_<drive letter>/<first node of include>

example:

z:/cnwsave/cnwtest/_drive_d/ExtraFiles would be the backup directory where environment cnwtest had specified a backup directory of z:/cnwsave and an include directory of d:/ExtraFiles.

NOTE : When a deployment automatically restores, if any include directories were backed up, these include directories will have file shares and cluster resources associated taken offline, all files within the directories removed and restored from the pre-deploy backup area.

Exclusion of Directories, Files, File Types

The following are automatically excluded from any managed backup (scheduled, ad-hoc, deployment):

- <root>\excludefrombackup – eg d:\example\excludefrombackup
- <system path>\backupinfo
- any file with file types of **tmp** or **cmp** in any directory being backed up
- any file with file type of **dat** when located in the system directory
- all files and sub-directories within the specified **JournalRootDirectory**
- all files and sub-directories within the specified **JournalArchiveDirectory**

Exclusions may be specified globally for all managed environments or for a specific managed environment.

Global exclusions are entered via the **Options | System Global Options** form in the **Backup** sheet.

Specific managed environment exclusions are added via the Managed Environment backup form. For a JADE environment, this is accessed via **Setup | JADE Environments** form via the **Backup** sheet and clicking on the **Schedule** button. For an ODBC environment, this is accessed by right clicking the database in the ODBC Maintenance form and selecting **Edit ODBC environment <name> backup specs**.

For backups of JADE Environments initiated by the deployment process, further exclusions may be defined in the ini file under section **[JSMDatabaseBackup]** with verb **DeployBackupExclusion<n>** where n can be from 1 to 99.

Example:

```
[JSMDatabaseBackup]
DeployBackupExclusion1=server\*.txt
DeployBackupExclusion2=server\c_system\testing\*.*
```

Exclusions are not to include the drive letter or the environment name as these are pre-pended at run time from JSA database values. I.e. the paths in the following table are relative to the database root directory. The following examples provide possible exclusion scenarios (where node is a subdirectory of the root node):

exclude directory	<node>*.*	Exclude files and all sub-directories within specified directory
exclude file type within directory	<node>*.txt	Excludes all file types of **.txt** within directory specified
exclude partial name	<node>\abc*.*	Any file starting with **abc** within directory specified
exclude partial name of file type	<node>\abc*.txt	any file starting with **abc** of file type **txt** within directory specified
exclude file name of any file type	<node>\abcxy.*	any file of title **abcxy** for directory specified for any file type
exclude file name for file type	<node>\abc.txt	File in specified directory named abc.txt

Backup Sentinel Files

Several sentinel files are used during a backup and are resident under the environment's **server/c_misc** directory unless otherwise specified.

backup.htm

Used by JSM, JSA and commander to detect whether a backup is in progress. This sentinel file is only resident under the active environment while a backup is in progress. NOTE: if a restore is done from a JSM Managed Backup, then the **backup.htm** file will need to be manually removed before commander can be used to start up the system.

_backup_offline.htm

Same duration as for **backup.htm** above, but provided only when the backup in-progress is an offline backup. This sentinel file allows non-standard batch scripts to detect the type of backup being run.

NOTE: if a restore is done from a JSM Managed Backup, then the **_backup_offline.htm** file will need to be manually removed.

_backup_OK.htm

This file is created in the backup area under the **server/c_misc** directory when the backup process has completed successfully.

_backup_requires_journal.htm

This file is created in the backup area under the **server/c_misc** directory when the backup process is online and for an SDS Secondary database. This signifies that the backup does not have all the transaction journals required to restore the database as other than an SDS Secondary database. The transaction journal required is identified in the contents of this file, this being the transaction journal number.

Successful Backup

When a backup is completed successfully, this is denoted by the creation of file **_backup_OK.htm** in the <default backup node>/<env>/server/c_misc and this file provides the following details (being version #4) with key items delimited by the <Tab> character:

- #4 <Tab> Backup Version <version>
- Time Completed (in format dd/mm/yy<Tab>hh:mm:ss)
- Host Environment Name
- Server Host FQDN
- Master Environment Name
- SDS Environment Id
- Backup Type - Offline or Online
- Database Role
- Database Subrole
- Commit Coherent – true (SDS Native backup can be restored to RPS) or false
- RPS Storage Mode
- Deployment Regime

- JADE Version
- JADE Character Set
- CardSchema Version
- Start Journal
- End Journal
- Offline Structures - <number of offline structures>
- multiple entries for each database directory :
  format 'Database_Directory<Tab><database directory>
- multiple entries for each backup directory :
  format 'Backup_Directory<Tab><backup directory>
- multiple entries for each **offline** database structure:
  format 'Offline<Tab><structure_name>
- multiple entries for each database structure:
  format 'Structure<Tab><structure_name><Tab><partitionID or space><Tab><source directory><Tab><backup directory><Tab><size (length 19 bytes with leading blanks)><Tab><size (in readable form)>

Note : Any record preceded by a colon (:) is treated as a comment entry.

Environment Shares

During an offline backup, any cluster file shares defined for the environment (and its sub-directories) will be taken offline and any open Window file shares will be closed. This also applies to any inclusion directories.

When the offline backup is completed, any cluster shares taken offline will be brought back online.

NOTE: The action of taking cluster file shares offline and online may generate SNMP traps which will shows as alerts within JSM.

Offline Backup Considerations

On a Windows platform, if the service is other than automatic and not a cluster service then following an offline backup the service will be left offline. On a Linux platform, if the service is not defined in any of the service run levels then following an offline backup the service will be left offline.

A managed offline backup is achieved by performing a quiesced online backup. When restoring an offline backup to a stand-alone system (one whose database role is undefined), the journal in the backup is not required and recovery is not required. If the backup journal(s) is used, then recovery is required (since in a quiesced backup, the journal advances during backup, even though user transactions are disallowed).
When restoring an offline backup to an SDS Primary, the journal(s) in the backup is required and recovery is required.

Backup Directory Permissions

The interim backup directory attempts to obtain the permissions from the final directory (if it is available).

If the final directory is not available, then the permission action is to Replace the permissions on the interim directory (see Replace action below).

If the final directory is available, the permissions defined there are applied to the interim backup directory replacing any that may have been inherited from the root node.

Then the permissions action as defined by the initialisation file is actioned. The permissions action is specified in section JSMDatabaseBackup with keyword BackupDirectoryPermissions. Valid actions are:

None	No changes to permissions
Edit	Add permissions below to existing permissions
Replace	Replace any permissions defined with those below.

If permissions are added or replaced then the following permissions will be set:

SYSTEM	Full Control
Administrators	Full Control
<environment>	Full Control
<environment>_G	Read access
<environment>_M	Read access

Backup Directory Shares

Shares defined for the directory and sub-directories are retained across a backup.

During the period where the now redundant backup is being deleted, any shares that were in use would have been forcibly closed down and any cluster shares defined would be taken offline

Backup Scripts

The following backup scripts only apply to scheduled or ad-hoc backups. They are deliberately not invoked during deployment related backups.

For these scripts to be executed, they must be resident under the environment's **<misc>\Cardinal** directory (for Windows this is **<env>\server\c_misc\cardinal** and for Linux this is **/opt/<env>/misc/cardinal**). These scripts are retained after execution (ie they are not renamed)

For details regarding the content of the scripts below, the variables items provided by JSM and for examples refer to section **Deployment Scripts** later in this document.

The <suffix> for the scripts in the table below is **cmd** for Windows and **sh** for Linux.

Script name	Backup Type	Description
prebackup_online.<suffix>	Online	Executes before the database files start being backed up
postbackup_online.<suffix>	Online	Executes after the last of the non-database files are backed up and after the JADE 5.0 media integrity checks have been run
prebackup_dbup.<suffix>	Offline	Executes before the database service is terminated. Note that this script will execute regardless of whether the database is online or not.
prebackup_dbdown.<suffix>	Offline	Executes after the database service has been terminated and the database is no longer active..
postbackup_dbdown.<suffix>	Offline	Executes after the backup has completed and the JADE 5.0 media integrity checks have been performed and prior to the database service being restored
postbackup_dbup.<suffix>	Offline	Executes after the backup is complete and the database service has been restored.

Housekeeping

JADE Environment

When the JADE environment has a log expiry days specified then the following directories with file types listed have those files deleted if they exceed the expiry days.

The following also applies for JSA environment housekeeping if it has been switched on (refer to JSA Housekeeping schedule in System Global Options)

Directory Use	Source	tmp	log	csv	rar / zip	dmp	err	txt
Transient Database	[JadeServer] TransientDbPath	Y						
Temp path	<logpath>/temp	Y						
Current Journals Path	<logpath>/current	Y	Y					
Archive Journal Path	<logpath>/archive	Y	Y					
Log Path	[PersistentDb] JournalRootDirectory	Y	Y	Y	Y	Y	Y	Y
JADE Activity Logs	[PersistentDb] ActivityLogDirectory	Y	Y	Y	Y	Y	Y	Y
CardSchema Logs, Stackdumps, Transaction Logs	[CardLog] LogPath	Y	Y	Y	Y	Y	Y	Y
Crash Logs	[FaultHandling] CrashLogDirectory	Y	Y	Y	Y	Y	Y	Y
Process Dumps	[FaultHandling] ProcessDumpDirectory	Y	Y	Y	Y	Y	Y	Y

NOTE : When JSM is backing itself up, the following addition file suffixes under the \logs directory will also be subject to housekeeping:

- .bat
- .end

ODBC Environment

When the ODBC environment or ODBC instance ('system' databases) has a log expiry days specified then the following directories with file types listed have those files deleted if they exceed the expiry days.

Directory Use	Source	tmp	log	csv	rar/zip	dmp	err
Log Path	JSM logs definition	Yes	Yes	Yes	Yes	Yes	Yes

Backup History

Each successful backup adds a single entry to a file 'backup_history.txt' which is maintained for each database environment under the environment's root directory.

The file is a comma-separated values file which can be useful to determine trends in growth and length of time to take backups The items recorded are as follows:

Start time
End time
Time when system was available
Elapsed.time
Backup Size
Date (CCYYMMDD format)
Database size
Non-database file size

Note that the last 2 entries are only available from when JSA 3.1 is deployed.

VersionInfo.txt

As part of each backup, the jverinfo utility is run at the start of each backup to ensure that the versioninfo.txt file is current as at the time of last backup.

This includes pre and post-deploy backups.

Misc File Copy Errors

A system wide option to allow backups to complete normally despite having one or more files from the Misc directory (or any of its sub-directories) fail to copy may now be invoked.

The default behaviour if this option is not turned on is for any backup to fail and immediately terminate whenever a file from the misc directory or any of its sub-directories fails to be copied.

This system-wide option is available from the **System Global Options** form on the Backup sheet and is a checkbox titled 'Continue backup process after misc file copy failures'.

Erroneous directories ignored

When the ini file specified JournalRootDirectory is on a different volume from the database, then the following directories are ignored if found on the database volume:

- <environment root>\logs*.*
- <environment root>\server\c_system\journals*.*

Should any of the following directories be located on the same volume as the database when the journal volume specified in the ini file is different, an alert is generated advising of the erroneous directory:

- <environment root>\logs\current
- <environment root>\logs\archive
- <environment root>\server\c_system\journals\current
- <environment root>\server\c_system\journals\archive

Successive Backup Failures

When 7 or more successive backups have failed to complete, an alert is now issued as shown in the following example

```
JSM Alert : Backup for tstx64 on cnwmfjvb.cnw.co.nz - multiple
successive failures

14/03/11 15:56:43 Backups for host environment TSTX64 of
CNWMFJVB has failed for the last <n> successive attempts.
Please investigate.
```

Managed Restores

This restore facility is provided both as a standalone restore mechanism and is also utilized by the deployment process when restoring environments for JADE 6 or later releases.

This section describes the process as a standalone restore mechanism.

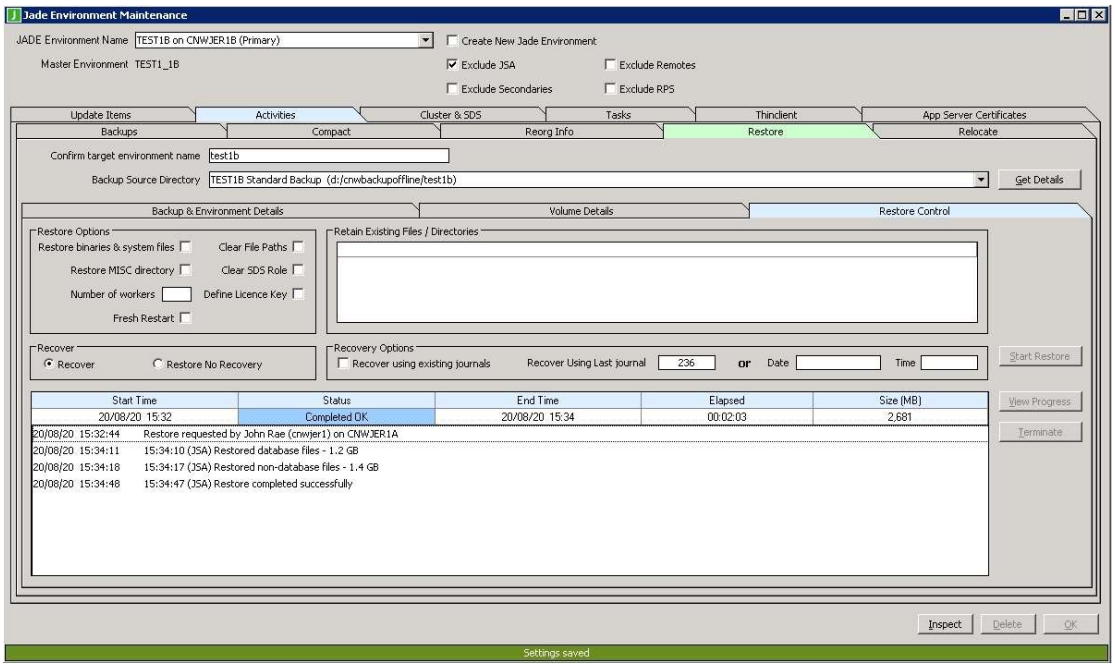
The restore mechanism may also be used to create / replace an environment from the backup staging area of some other environment.

Restrictions

SDS environments may only be restored from the pre-installation or standard backup staging areas and no journal information is required as resynchronization will occur when re-connection to the SDS Primary occurs.

Initiation of a Restore

The restore facility is started from the host environment maintenance form. The JADE environment being restored is the host environment selected on this form.



Selecting the restore sheet on this form presents the above details:

- the restore directory combo box is loaded with the standard and pre-installation backup directories for all other JADE environments resident of the same server host and is pre-filled to specify this environment's backup directory.
- The JADE version combo box is loaded with all valid JADE releases from the product release directory maintained in JSM. This ensures that the Jade version you are restoring is the Jade version you expected. If not, then the restore will fail in JSA during start up.
- The 'Retain Existing Files / Directories' is obtained from the Exclusions list for this environment and may be modified.
- The 'Recover using existing journals' is set by default. This means that journals from this JADE environment are used rather than the journals from the backup directory.

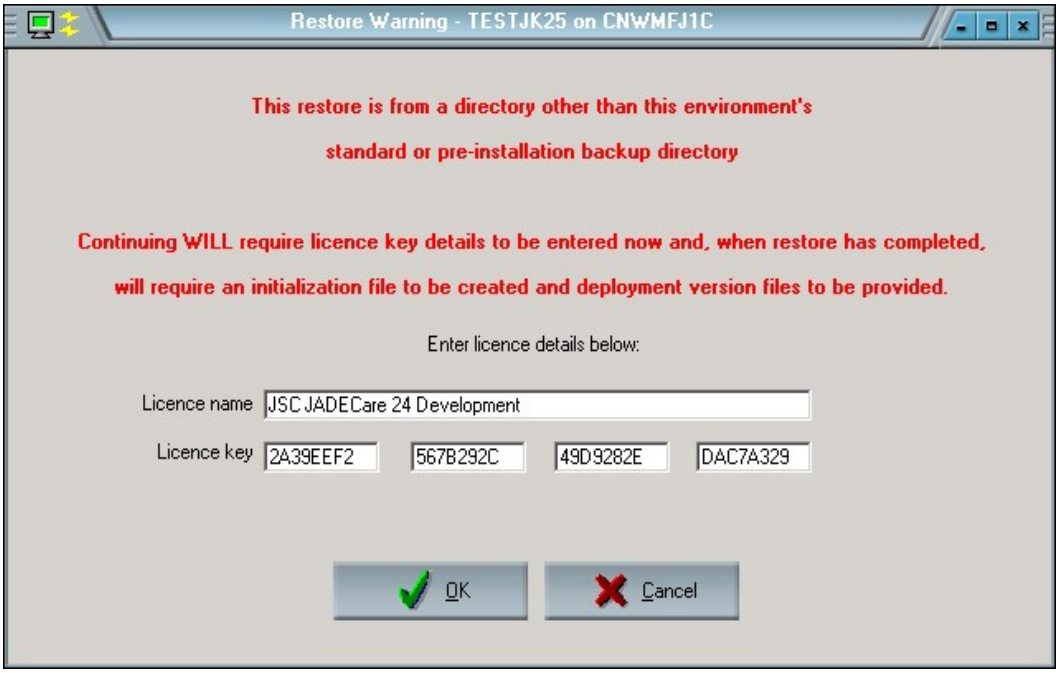
- Number of workers is pre-filled using the same value as defined for the backup offline workers. This value may also be adjusted before initiating the restore.
- Details of the last restore are shown if a managed restore has been initiated.

The backup directory specified for a restore may be one of those pre-defined in the 'Restore Directory' combo box or may be a manual entry.

Unless 'Restore No Recovery' is selected, one of the recovery options below must be defined:

- Recover to last journal, a journal number must be entered.
- Recover to specified date and time.

Clicking on the 'Start Restore' button will display the following form if the restore directory specified is not the pre-installation or standard backup directory for this environment.



If the JADE environment being restored has previously been an active JADE system, then the last known licence details and serial are pre-entered and need to be confirmed.

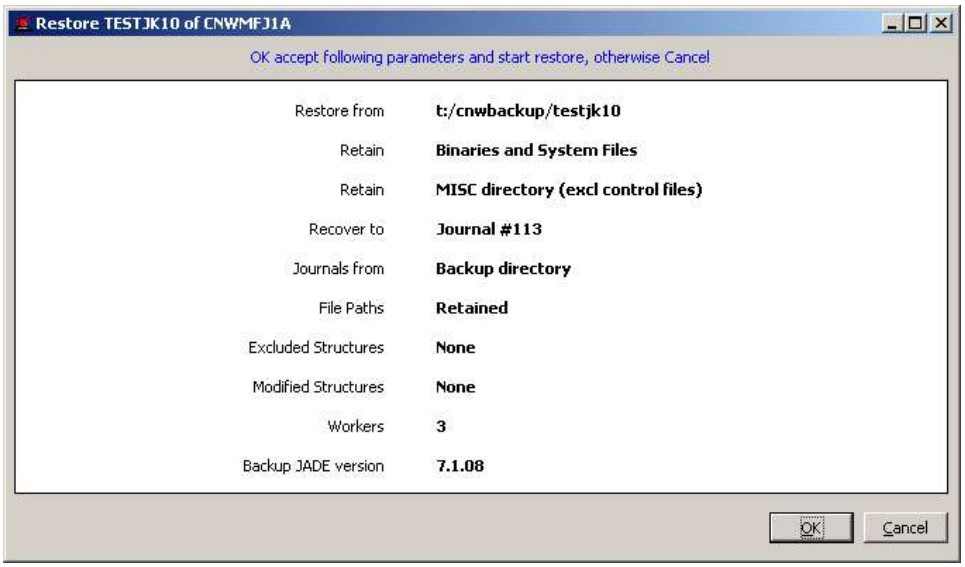
If the JADE environment being restored is a new environment and has never been run as a managed JADE environment, then no licence details will be shown and these details will need to be manually entered.

The licence details are validated before proceeding, any error will result in this form being re-displayed.

NOTE: If the JADE environment is being 'refreshed' from other than it's own backup directories, the restored environment will be left in a usable state, however, there will be no initialization files, JAWS control files or deployment version files resident. These will require operator intervention to provide once the restore has completed.

Start Restore

Having clicked on the start restore button and possibly providing the licence details, a final check form is provided to ensure that the parameters selected are correct as follows:

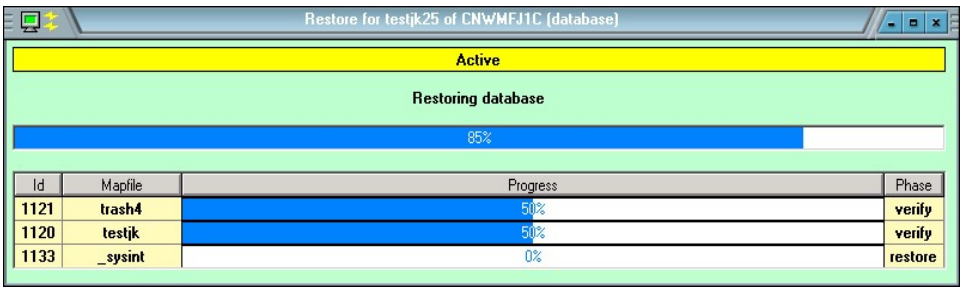


Check the parameters are correct and click OK to proceed with the restore or Cancel to modify the parameters.

Once the restore parameters have been verified and OK'ed, JSA is requested to start the restore process.

View Progress

A progress form may be activated from JSM so that the progress of the restore may be observed. It works in the same manner as the backup progress bar.



While in the database restore phase, the id shown reflects the process instance id seen in the System View form for the worker application. The phase column may have the following values:

- Restore – the mapfile named is being restored from backup
- Verify – the mapfile named has been restored and the integrity of the restored file is being checked.

Terminate Restore

This option is only valid if the restore log is:

- Active (currently in progress and JSA is online);

- The restore is a managed restore rather than an deployment process restore.

When this option is selected, a pop-up dialog is provided asking you to confirm your request before passing an interrupt request to the restore process concerned.

NOTE: The interrupt request may take a while to action as it may arrive just after a large mapfile restore request has been started or a large binary file has started being copied. If this is the case, then the process concerned will need to complete normally before the interrupt can be actioned.

Restore Processes

Once JSA has started the restore, the restore is completely autonomous from JSM. Regular updates are returned to JSM so that progress may be observed (via the restore log updated on the Host Environment form restore sheet or the Restore Progress form).

When the restore process has ascertained that the restore is viable, then JSA will place a backup.htm sentinel file in the <env>/server/c_misc directory to:

- Avoid backup and/or deployments from touching this environment
- Prevent JAWS from starting this JADE environment.

Any files found in the logs directory of the JADE environment being restored are not deleted. All existing journals found in the current or archive directories are renamed with a save_ prefix for potential re-use should the restore fail or any other issues arise with the system.

The restore process is performed completely by JSA using the following JSA applications (which will be visible via the System View screen), each application is suffixed by the JADE environment being restored:

KarmaRestoreMgr_<env>:

The parent application for any restore. This application starts and monitors all other worker applications, does all the logging required and periodically issues updates to JSM of progress to-date.

KarmaRestoreFile_<env>:

The name of the worker threads started for each mapfile to be restored.

KarmaRestoreCheck_<env>:

The name of the worker threads started for each mapfile to be verified.

Each restore creates a restore log (in the same fashion as the installation log and backup log). This is retained under the environment's logs directory and has the following title format:

r_<env>_<ccyymmddhhmmss>.log

Restore and Directory Structure and Permissions

The target environment's directory structure is retained, if it exists, and permissions applied will be done in edit mode retaining other existing permissions. Permissions specified below will have their access rights replaced and all other permissions are retained:

- SYSTEM – Full Control
- <administrator group – derived> - Full Control
- <env> - Full Control

- <env>_g – Read Access
- <env>_m – Full Control

The following steps are now taken during a restore:

- Cluster file share resources are taken offline for the JADE system's environment and for each include directory specified. NOTE :this could result in SNMP traps being generated.
- Open file shares are closed for the JADE system's environment and for each include directory specified.
- All files are removed from the JADE system's environment and for each include directory specified. Exceptions to this removal are file and directory exclusions and the transaction journal directories. The directory structure is retained to retain any permissions and file shares defined.
- All binaries and miscellaneous files for the JADE system's environment are restored from the pre-deploy backup area.
- All files from include directories and sub-directories are restored from the pre-deploy backup area. (refer to **Include External Directories** in the section **Managed Backups** above).
- Cluster file share resources are marked as online.

NOTE: The action of taking cluster file shares offline and online may generate SNMP traps which will shows as alerts within JSM.

Database Only Restore

If the options 'Restore binaries &system files' and 'Restore MISC directory' are not selected, indicating a database only restore, then if the backup area contains directory **<env backup root>\server\c _misc\apps** then this directory and its contents (including sub-directories) are restored unconditionally.

Note that the corresponding files and directories from **<env backup root>\altos\<characterset>\x64-msoft-win64\misc\apps** will also be restored to ensure consistency and avoid downstream deployment issues.

Files Automatically Excluded

If the restore is from a backup directory other than the environment's own pre-installation directory or standard backup directory then not only are those directories specified on the restore initiation form excluded from the restore but the following files are also excluded from being copied back:

- 'pem' file extension and binary path (<env>/server/c_bin)
- 'ini' file extension and binary path (<env>/server/c_bin)
- 'ctl' file extension and Cardinal miscellaneous path (<env>/server/c_misc/Cardinal)
- 'cmd' file extension and Cardinal miscellaneous path (<env>/server/c_misc/Cardinal)
- 'ver' file extension regardless of path
- 'exe' file extension where file is prefixed '<env>_cl_', '<env>_tc_' or '<env>_jom_' (these files all being deployment self-extracting files)

Ini File Specified Exclusions

It is now possible to add exclusions of files that are NOT to be overwritten by the restore process without patching JSA. This can be done by adding one or more ExcludeFile<n> entries in new section [JSMRestore]. Refer to ini file documentation earlier on section [JSMRestore].

Example:

An example is where a Jade Environment has .Net applications (running under one or more sub-directories in \server\c_misc\apps) and one or more requires a new configuration file named 'appsettings.json'. The ini file settings below would ensure that the appsettings.json in the active directories are not overwritten by the restore process.

```
[JSMRestore]
ExcludeFile1=d:\example\server\c_misc\apps\*,false,appsettings.json
```

The "ExcludeFile1' definition would ensure that any files named 'appsettings.json' json' found in the first level of sub-directories in d:\example\server\c_misc\apps are retained and not overwritten by files in the backup image.

Execute Operations scripts During Restore

If the any of the following command batch scripts are present in JADE database environment '\server\c_misc\Cardinal' directory they will be executed in the defined restore phase.

If any script terminates with exit code other than 0 then the restore will fail with error.

Script name	Execution phase
restore_initialize.cmd	Before any restore activity started
restore_post_remove.cmd	After existing file content has been removed and before database files are 'restored'
restore_post_dbcopy.cmd	After database files have been restored to disk and before any non-database files are copied back
restore_post_filecopy.cmd	After non-database files have been copied back from backup staging area and before database is recovered. This will execute if present even if non-database files are expressly not copied back.
restore_post_recovery.cmd	After database has been recovered and before restore process terminates

Script Maintenance

Script Maintenance provides the ability to view and modify (replace) scripts within JSM.

As at the 4.1.02 release, scripts can be executed in three contexts (three Script Activities):

- ODBC Backup
- Task Scheduling
- Webhook Endpoint

All scripts loaded currently will belong to one of these activities.

There is currently only one script defined for the ODBC Backup activity and that is to support the backup of the MSSQL environments (including system databases master, model and msdb).

Task Scheduling activity supports scheduled execution of PowerShell, DOS batch (.bat and .cmd) and ODBC query scripts (.sql).

Webhook Endpoint scripts can only be created by using the Script Builder form.

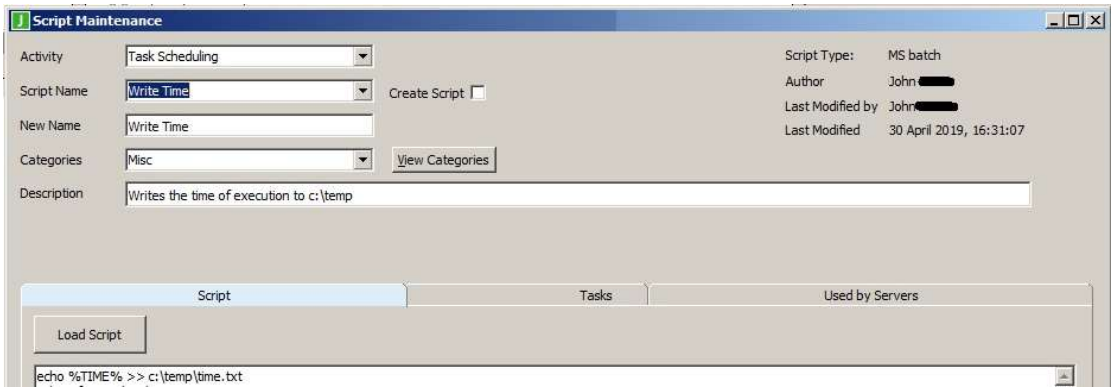
Creating a New Script

Webhook Endpoint Scripts

Scripting for webhook payloads is very specific. Please refer to the section titled 'Webhook Endpoint Processing'.

ODBC Backup and Task Scheduling Scripts

Open the Script Maintenance form (by selecting menu item Set Up | Scripts) and, when the form is presented, click the check box 'Create Script'.



NOTE: The 'Create Script' check box is NOT provided when the activity is ODBC Backup as the scripts required are under the control of the JSM development team. However the content may be modified at any time by the Operations Group.

In the script name define a unique name that the script is to be known by. This script name can be modified later if required.

Click on the 'Load Script' button to pop-up a file browser window and locate a valid script file, this can be any file with the following file types:

- bat – MS-DOS batch script
- cmd – MS_DOS command script
- ps1 – Powershell script

- sql – SQL query

The content of the script identified is loaded into the text box for viewing and the OK button will save this script.

ODBC Query Scripts

ODBC query scripts can be created in all JSM instances, however can only be executed via the task scheduling mechanism in JADECare24 JSM.

Note: ODBC queries that select a 'name' property must rename that column heading using an 'AS' keyword. E.g. SELECT name AS productName FROM…

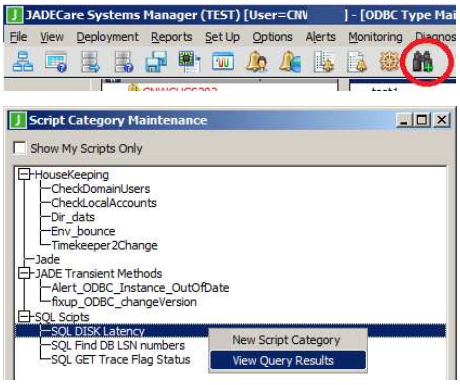
ODBC query scripts can be executed via the task scheduling mechanism against user and system databases (for example master, model and msdb databases in MSSQL).

cnwsql Query Limits

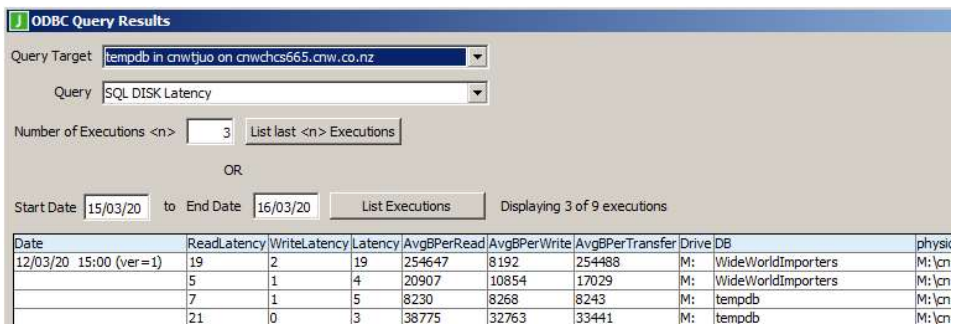
Maximum SQL statement result buffer size	48,000 bytes
Maximum SQL statement result rows	4096
Maximum SQL statement result row size	4096 bytes
Maximum SQL statement result column width	256 bytes. (right truncated)

Viewing ODBC Query Results

By default, output from ODBC query scripts is persisted and can be viewed (and deleted) from the ODBC Query Response form (via the quick launch menu in System view, or by right clicking the script in Script Category maintenance).



Query Results can be displayed by selecting the number of executions that you wish to view, or by specify a data range withing which the results were obtained.



JADE Transient Method Scripts

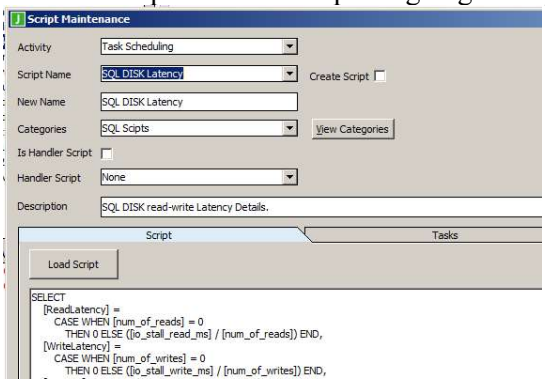
JADE transient methods can be executed through the Task Scheduling mechanism either as stand-alone scripts (for example reporting) or as partner scripts that process output from their counterparts. JADE transient method scripts can only be created by members of the JSM development team.

Task Scheduling Partner Scripts

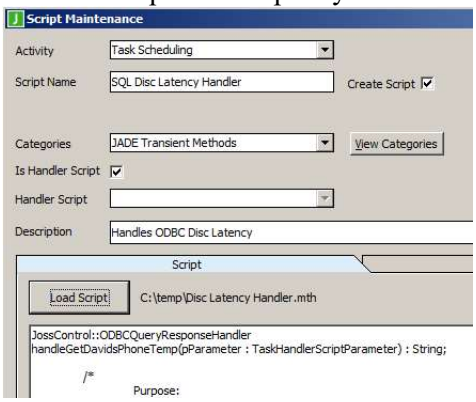
Partner scripts are JADE transient methods authored by the JSM development team that process output from their counterpart script. This provides the ability to define and modify script processing between JSM releases (as opposed to the default behaviour of merely logging the output in a task log).

Steps

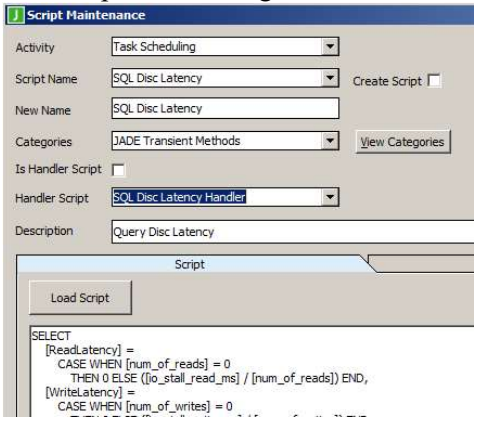
1) Create a script s1 whose output is going to be processed



2) Create a script s2 and specify that it is a handler script

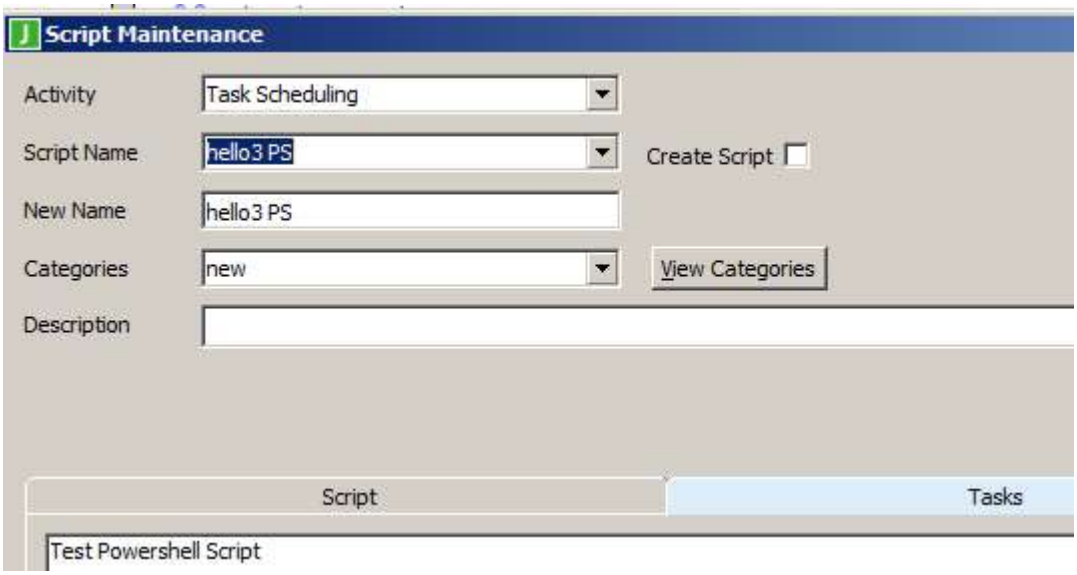


3) Edit script s1 and assign s2 to be its handler script



Identify Tasks and Servers Using a Script

The 'Tasks' and "Used by Server" sheets list those Tasks and servers that use the selected script respectively. Double-clicking on a row that defines a Task will cause the Task Maintenance form for that task to be shown.



Script Categories

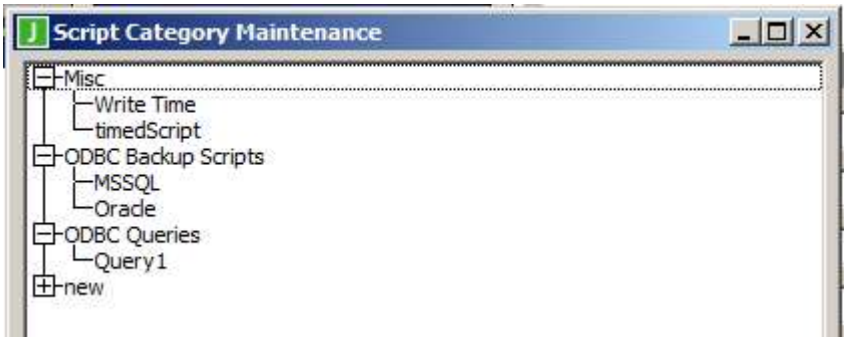
This form enables scripts to be grouped into user defined categories for easier viewing and management. The form is accessible from the main Setup menu and from the Script Maintenance form.

The form provides easy access to editing a script by double clicking that script to edit it in the script Maintenance form.

Initial categories "Misc" and "ODBC Backup Scripts" are provided.

Right click in the categories list to add a new category.

A script can be dragged from one category to another.



Task Scheduling

A task can be created/updated that will execute either a script or a command line sequence and this task can be targetted to one or more servers, JADE Environments or ODBC Environments.

It is possible to schedule tasks for any of three managed entity types (target types):

- Server
- JADE Environment (database or remote fat-client)

- ODBC Environment

A task defined to execute with a specific target type is restricted to executing only for members of that target type.

When a script executes against an instance of a target type, variables pertaining to that target type are inserted into the start of the script prior to execution. The values assigned to the variables are determined by the target instance. The author of the script may include references to those variables in the body of the script. Note: for a task that runs against a server, the variables inserted are those for the Jade environment that is the JSA.

Any standard output received from the process or processes started by this task may be saved to the history log for the executed schedule – see subsection "Standard Output Handling" below.

A task defined with a target type of JADE Environment can be executed for a number of different JADE Environments which may all exist on the same server or each may be on different servers.

A task may be defined with the same schedule for each target requested but it is also possible for each task target to have totally different schedule. A target may have one or many schedule definitions.

WARNING: When starting any batch process, be aware that all child processes spawned will inherit the standard output handle (unless this is expressly avoided). That means that what may be considered an asynchronous task turns out not to be and the task schedule run may execute for considerably longer than expected as it is waiting for the standard output handle to be relinquished.

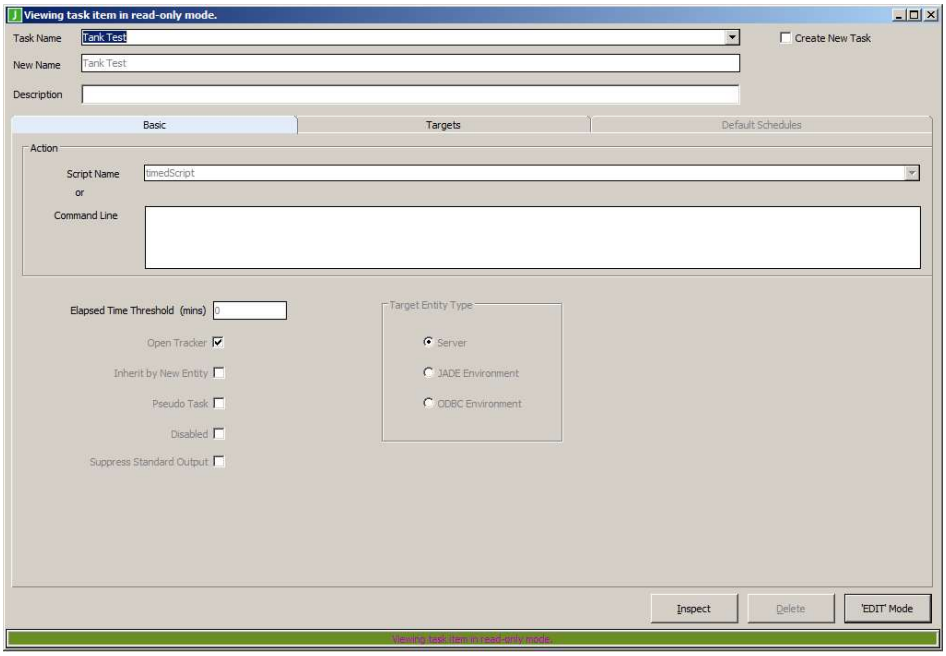
Items to be aware of:

- All of the changes made in the Task Maintenance form are only committed to the database and replicated to JSA when the OK button is clicked (so "Run Now" and "One Time" schedules can expire before the update is committed).
- All changes will be shown by colour selections to indicate:
    - Green - added
    - Orange - changed
    - Red – deleted
    - Black – no change

The Task Maintenance form can be displayed from the following sources:

- Menu options (Set Up | Task Scheduling)
- Server Host Maintenance form sheet Task – double click on a row
- JADE Environment Maintenance form sheet Task – double click on a row
- ODBC Maintenance form – Task Item list – double click on a row
- Task Summary form – double click on a row

Creating a New Task



Click on the check box 'Create New Task' to create a new task.

Fill in the items on the 'Basic' sheet:

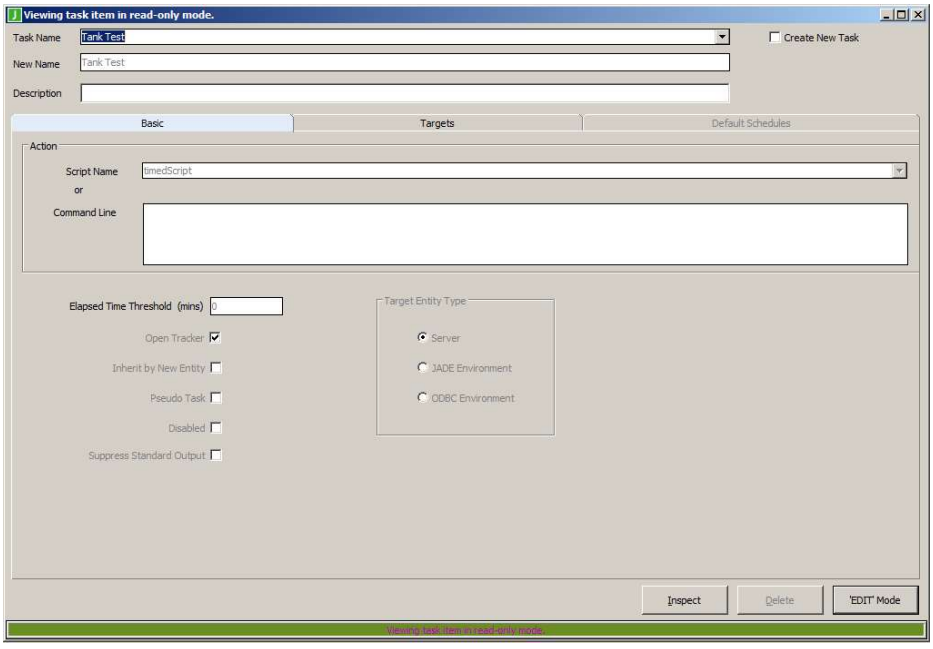
Task Name	A unique name that identifies the function. This can be changed later if required.
Script Name	Select a script if the 'command line' option is not to be used
Command Line	Define the command line syntax if a script is not to be used.
Elapsed Time Threshold (mins)	Define the 'elapsed time' in minutes that the task is expected to complete in. If this threshold is exceeded then a Tracker call or an alert will be created / updated. If left as zero, then no threshold check is done. NOTE: this threshold will be inherited by each target entity. It can be overridden for specified targets.
Open Tracker	This check box is only available if Tracker has been instantiated. Check this box if Tracker calls are to be raised with any threshold or failure relating to this task. If this box is not checked or Tracker is not instantiated then alerts will be generated for any issues arising from this task item.
Inherit by New Entity	Default schedules will be inherited by any new entity created of the designated target type. For example: If a new JADE Environment is

	discovered then the new entity will automatically acquire the task items which: - have this attribute set and - where the defined target type is JADE Environment The JADE Environment will have the default schedule definition(s) as defined for the task item.
Pseudo Task	Feature yet to be implemented.
Disabled	When set, this means that all scheduled entries for all target entities no longer execute.
Suppress Standard Output	Script 'info' lines are not captured (and not saved to JSM's history log). This enables standard output from any subordinate processes started by the task script to be captured and processed by the task script.
Target Entity Type	Select one of Server, JADE Environment, ODBC Environment. This restricts the selection of Task Targets to this specific type and determines the variables (and their values) inserted into the script prior to execution.

Updating an Existing Task

Allows for any or all of the following:

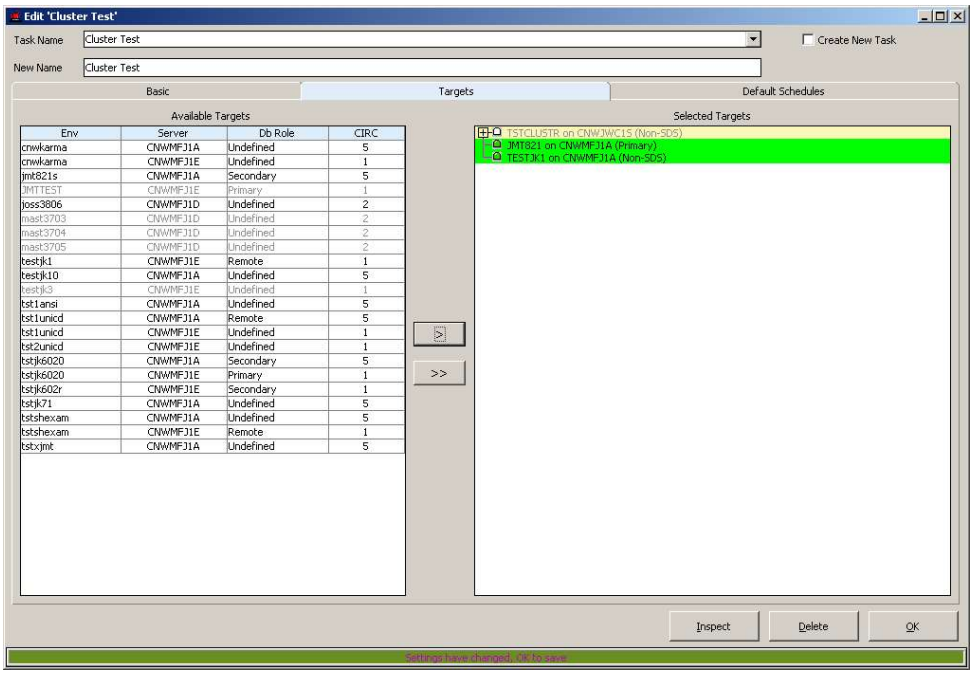
- the task name to be changed
- for task item specifics to be changed (e.g. elapsed time threshold)
- task targets to be added or deleted
- schedules to be added, changed or deleted
- default schedule items to be added, changed or deleted.



Selection of Task Targets

The available task targets will be dependent on the Target Entity Type specified. These are displayed in a table with the details varying dependent on the target type. Note that the table may be sorted by clicking on the column heading.

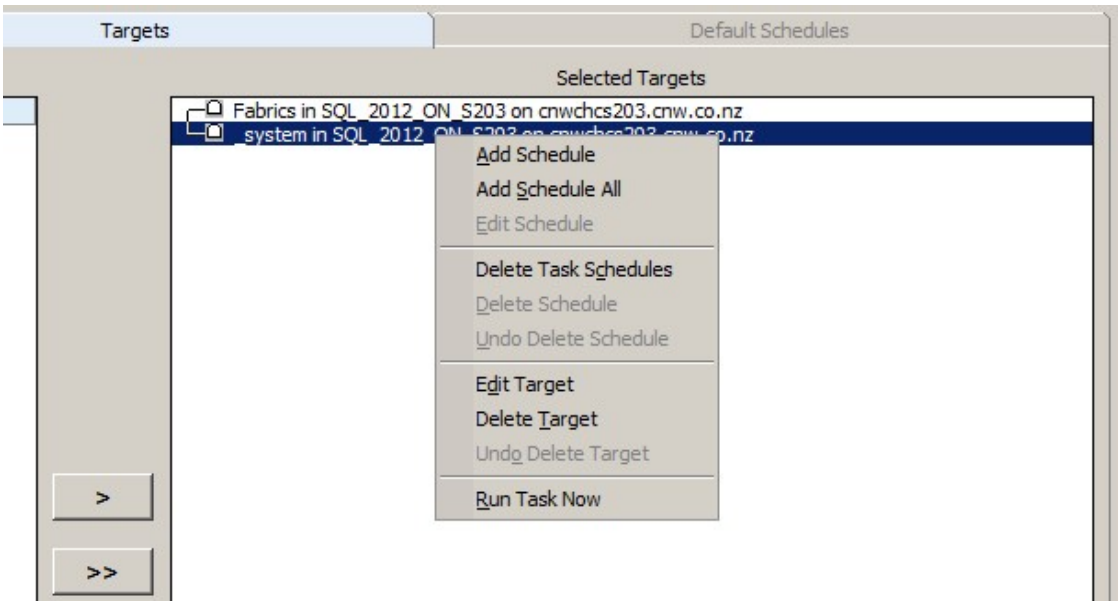
The following form lists of available and selected servers for a task having target type server (just the short FQDN is displayed):



The above shows three server targets selected (where the new targets are highlighted in green).

Definition of Schedule Details

By right clicking on an entity in the 'Selected Targets' area, the following menu options are provided:



Select menu option "Run Task Now" to create a schedule for immediate execution. The schedule added is is for a time in the time frame of the target server within the next thirty seconds to two minutes.

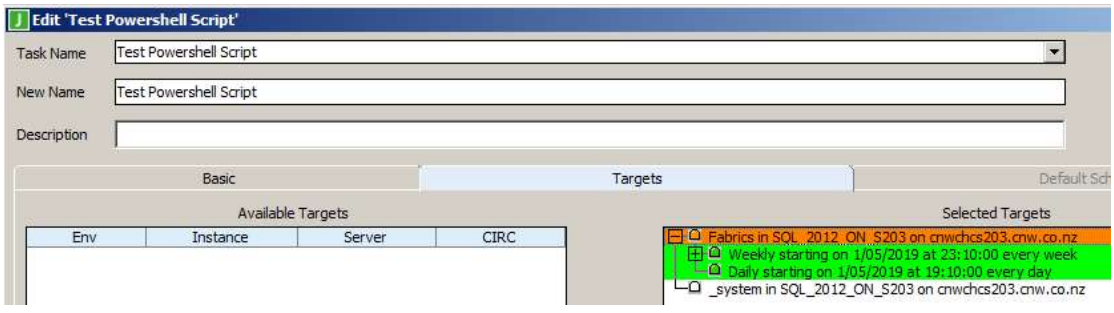
Select menu option 'Add Schedule' to create a schedule for the task target selected.

Select menu option 'Add Schedule All' to create a schedule for all task targets currently shown in the 'Selected Targets' area.

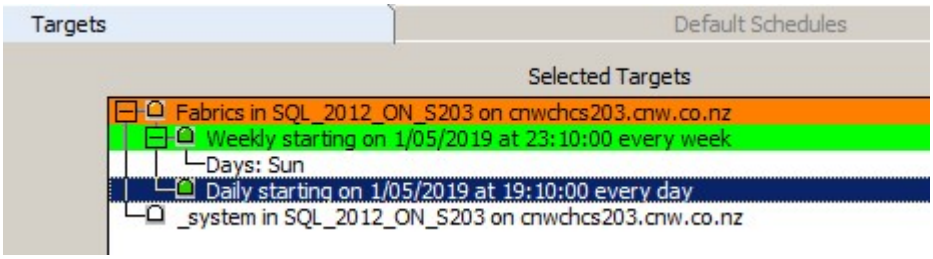
The schedule dialog that appears will identify in its heading caption the selection made:

- Add schedules to <target name>
- Add schedules to multiple targets

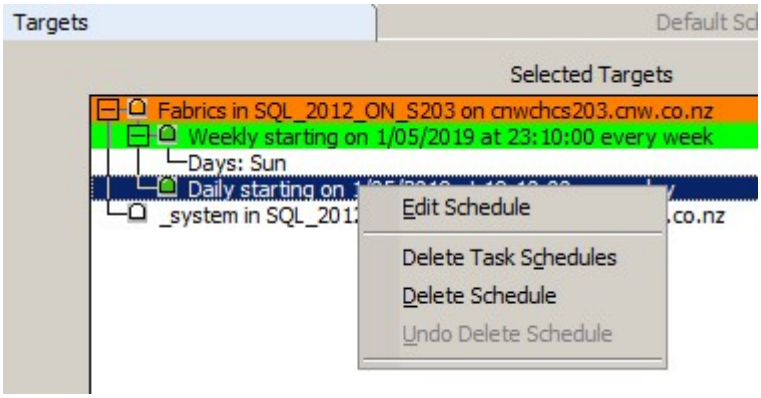
When one or more schedules have been defined for a task target, a + sign is displayed next to the target.



Click the + sign to expand the schedule details as follows:



Right clicking on a schedule pops-up the following dialog:



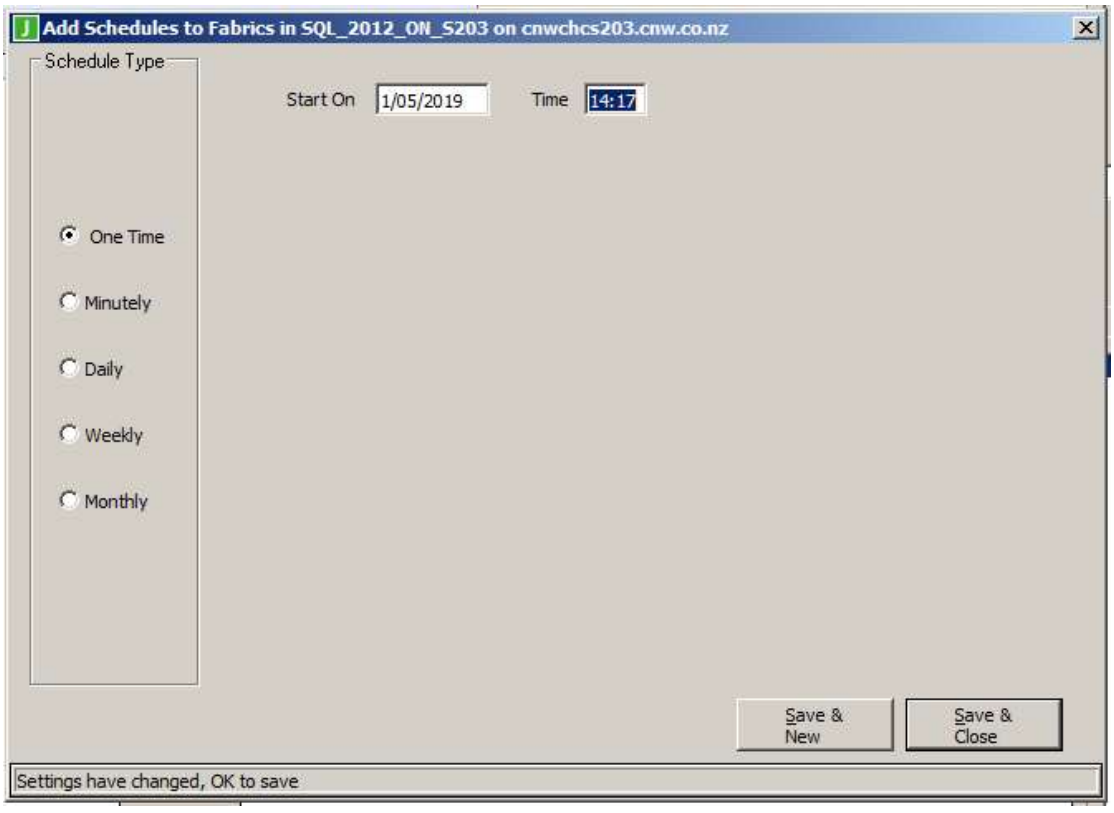
This allows the schedule to be edited or deleted.

Schedule Definitions

The scheduling options are similar to those provided by the Windows Task Scheduling with some extra options. An example of each schedule option permitted is listed below:

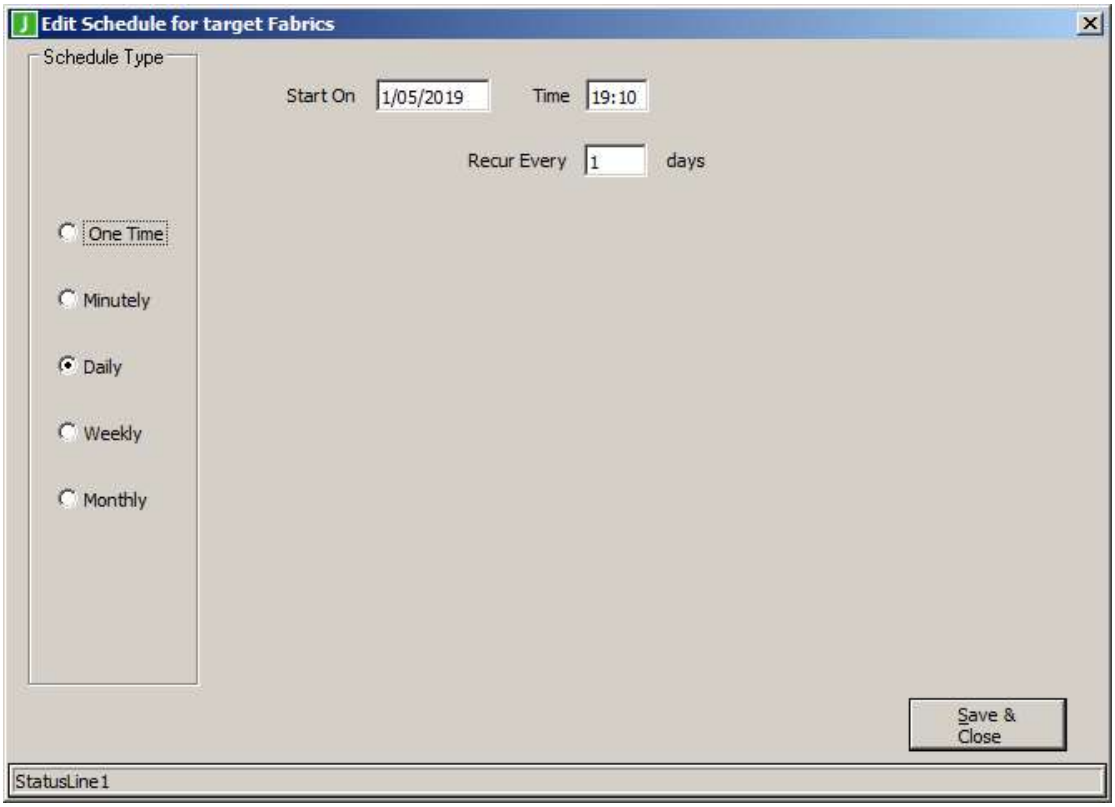
One Time Schedule

Provides the ability to define a single run on a specified date and time.



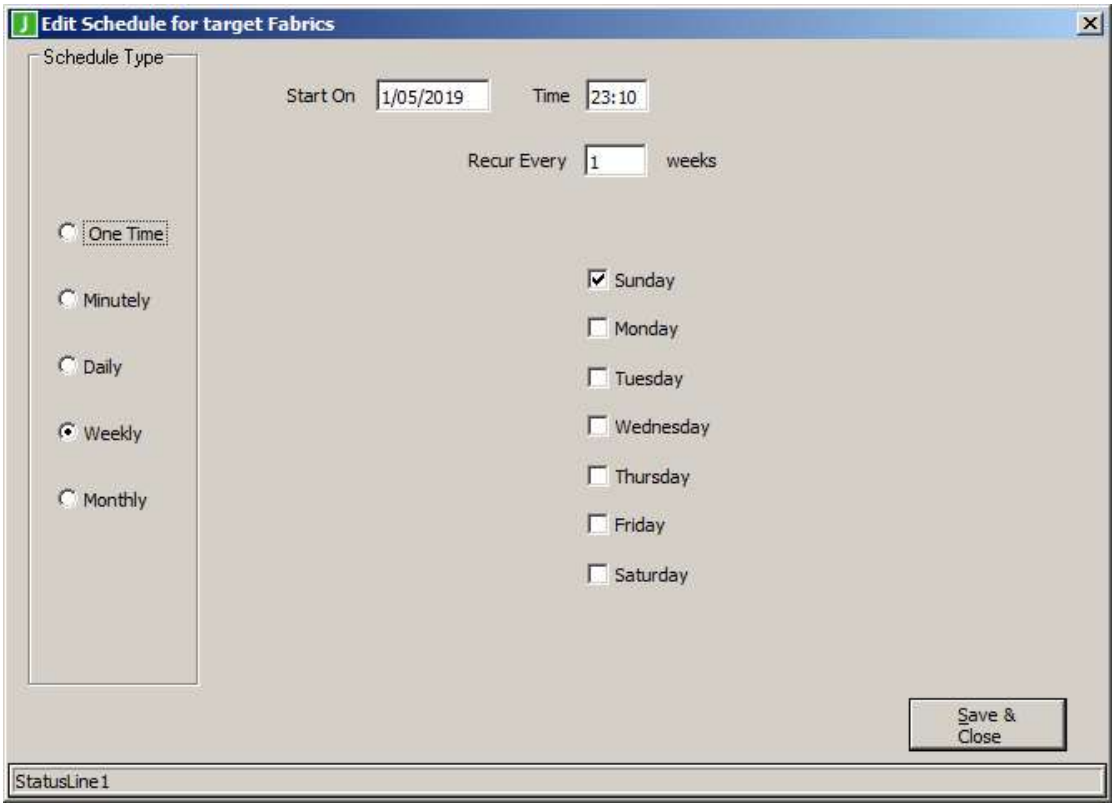
Daily Schedule

Provides the ability to run every specified number of days starting on a specified date and time:



Weekly Schedule

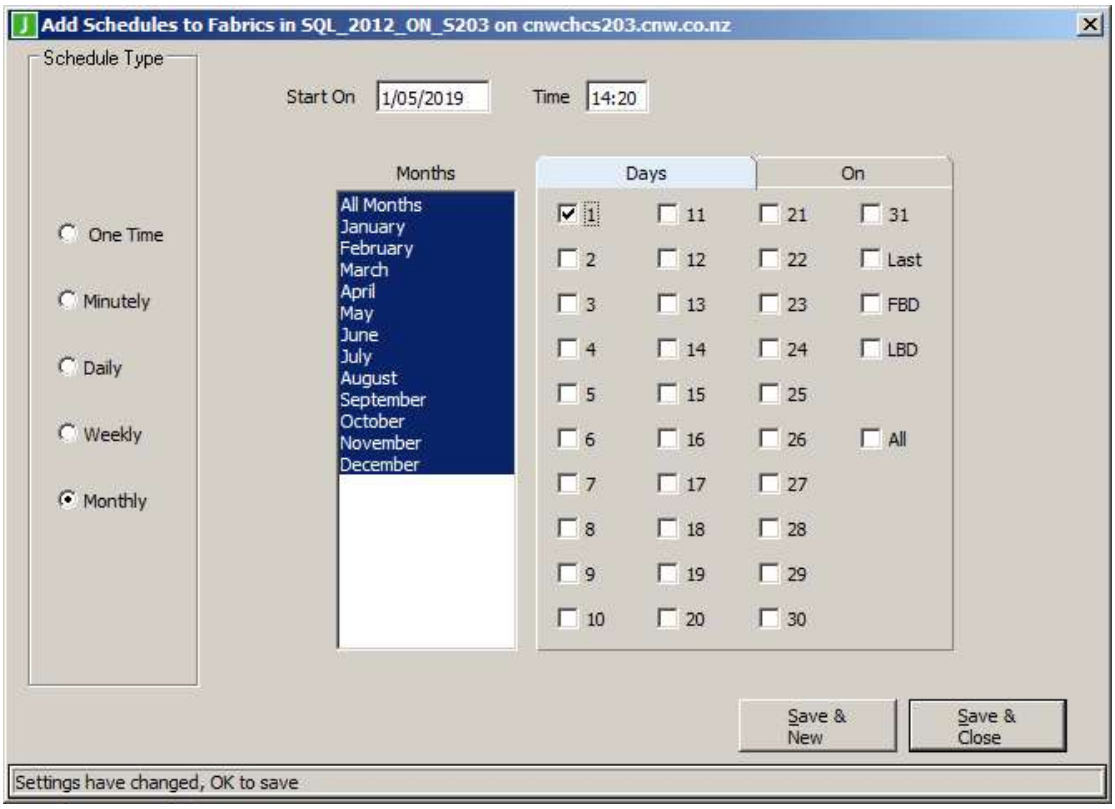
Provides the ability to run on specific days of the week and the weekly intervals starting on a specified date and time:



Monthly Schedule by Day of Month

Provides the ability to:

- Run for every month of the year or for specific months of the year
- Specify which days of the month(s) selected the task is to execute for.

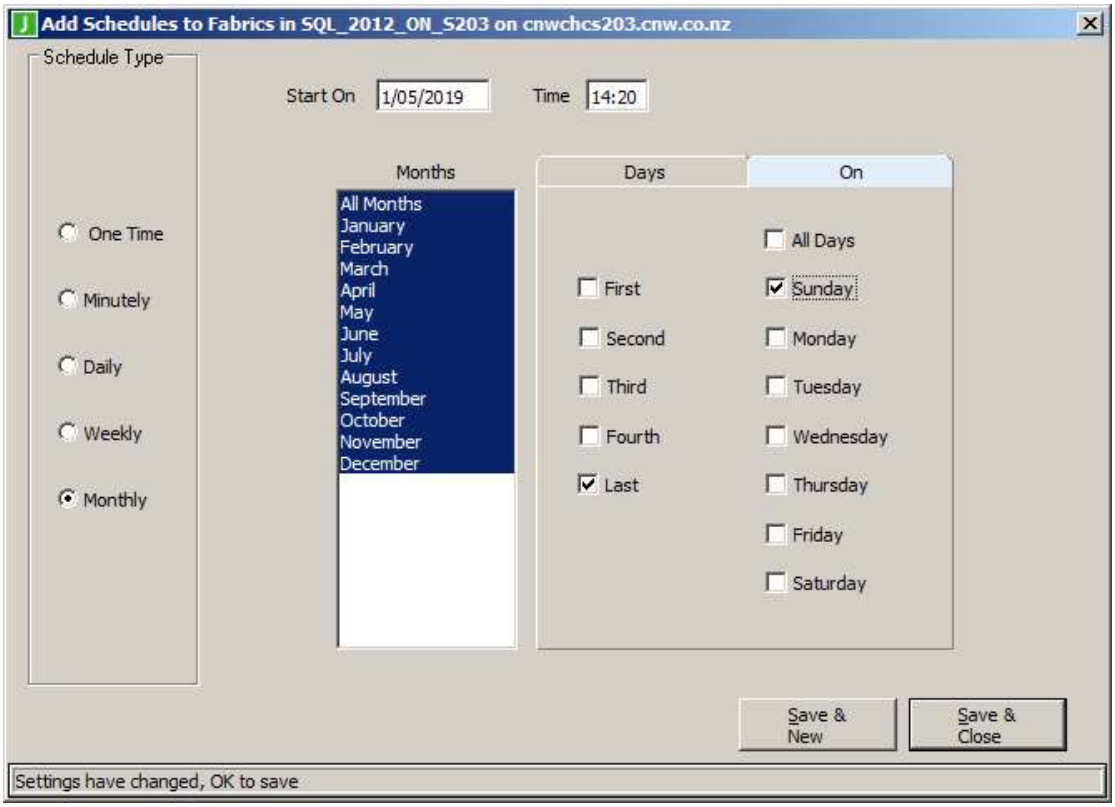


Last	means the last day of the month. So specifying last in the month of February would mean either the 28th or 29th (if a leap year)
FBD	means the first business day of the month.
LBD	means the last business day of the month.
All	When checked will set every day to true. When unchecked will reset every day to false

Monthly Schedule by Day of Week

Provides the ability to:

- Run for every month of the year or for specific months of the year
- Specify the week or weeks of the months selected
- Specify the day or days of the weeks selected.

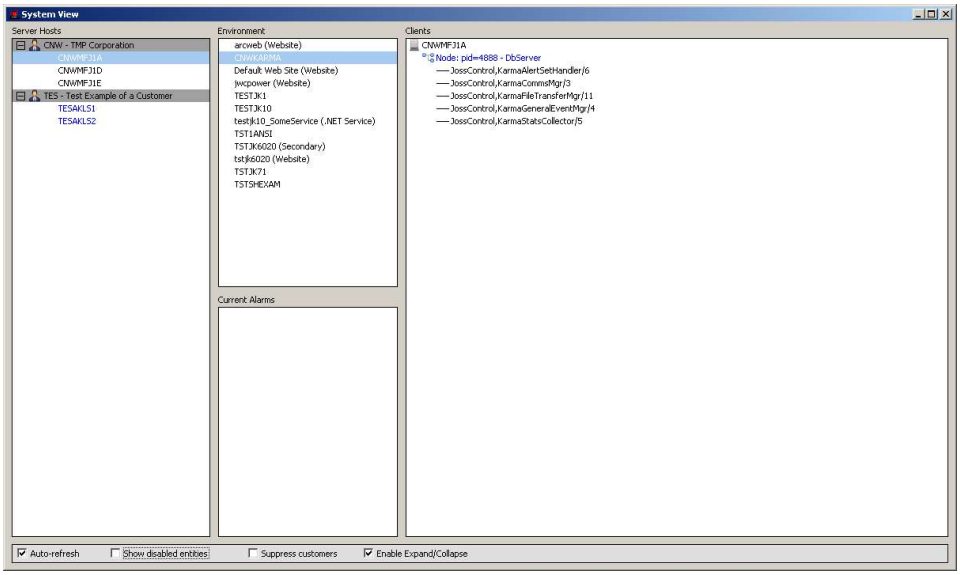


Standard Output Handling

By default, any standard output received from the process or processes started by this task (for example, a batch script) is passed back to the initiating JSM\JSA application (unless Suppress Standard Output is selected on the Basic sheet for the task). Each line of output is logged to the task log (t_log). The first word of each output line is checked to see if it matches one of the following command words and the corresponding command action is performed.

Command	Command Action
"taskstarting"	Signals that any following ‘information lines’ (see next command) will be sent to JSM and stored persistently in a history log. Any “info” lines prior to this will be logged in the t_log but will not be saved in the JSM history log.
“info”	Text following the “info” command will be saved in the history log.
“info:”	“info” with a colon. Synonym command for “info”
“error”	Generates a “Script : error string” entry in the t_log with the text that follows the “error” command
"taskcomplete"	“Task <task name> "script execution completed." is written to the t_log.

System View



Double-clicking on any entity within this form (with the exception of Current Alarms) will open a new dialog showing further details for the entity concerned.

Auto-Refresh is ON / OFF

This check-box provides for the form to be refreshed as details change. The default value for this check-box is off as there can be quite a number of notifications received by this form when this option is selected.

The check box value is retained across a form close and open for the user.

Show Disabled Entities

When the above option is enabled, then all entities (server hosts and host environments) marked as disabled can be viewed via this form.

When the option is disabled, then disabled entities are suppressed from view.

The check box value is retained across a form close and open for the user.

Enable Expand/Collapse

This is only visible for users defined with security roles of Super User or Operator.

This option will allow entities in the Server Hosts list box content to be collapsed or expanded. When collapsed this will reduce the number of items visible in this list box.

The check box value and the items that are collapsed are retained across the form close and open for the user.

Cluster Host

Right click on a cluster host will select the cluster host concerned and display a menu with the following options:

- Cluster host details for <cluster host name>
- Delete cluster host <cluster host name>
  NOTE that this is the only way to now delete a cluster host.

Server Host

Placing the mouse over a server host will cause a bubble help dialog to display showing the date and time local to the server host concerned.

Right click on a server host will select the host concerned and display a menu with the following options:

- Windows Terminal Server session to <host name>
- Windows Terminal Server console to <host name>
- RIB Connect to <host name>
- Commander session to <host name>
- Other remote session …..
- Host setup for <host name>
- Host details for <host name>
- View Backup Schedule
- Create HostDown call for <host name> or Open HostDown call <xxx> for <host name>
- Discover .NET Services
- Discover Web Info
- Delete host <host name>

The above may be enabled / disabled depending on the Operating System detected on the host concerned.

The first 5 options provide access to various interfaces that provide access to the host concerned.

The next 3 items permit the user to go directly to the appropriate JSM form and have the correct details presented for the host selected.

The 'Create HostDown call / Open HostDown call" is only available if the Tracker module has been installed.

The menu items 'Discover .NET Services' and 'Discover Web Info' will:

- only be visible if the JSM global Discover option is enabled and
- enabled if the server host 'Enable Discovery' is set

Host Environments

Right click on a server (host environment) and the item will be marked as selected and a menu provided:

- Host environment Setup
- Host environment details
- Show Host Environment Call <xxx>
- Start Adhoc Backup
- View Backup Progress
- View Backup Schedule
- Delete Host Environment <host env> of <host name>
- Change Role to SDS Secondary

"Show Host Environment Call" is only available when Tracker has been installed and a server host down fault has been exploded to generate sub-calls for all host environments that run on the server host.

"View Backup Progress" is only valid while a backup is actually running.

"Delete Host Environment" is only valid when the host environment is not active.

"Change Role to SDS Secondary" is only visible when the selected environment is an SDS Primary and is enabled when the database is NOT active and the SDE has more than one SDS Primary defined.

Clients

This gives a display of the JADE applications that are executing, and where running as a thin-client, the computer name of the thin-client workstation.

The applications are ordered by server host they are running on, then the JADE node O/S pid, and then by the JADE process instance id.

Where the client node is an Application Server Node and the environment has deployed JADECare Start 3.5.01, then the node text may be appended with the literal '(SSL)' or the text may be coloured red. With JADECare Start 3.5.01 or later, it is expected that application servers should be using SSL and those application servers that are not are highlighted by the red text.

Server Host Fault Explosion

This feature is only available if the Tracker module has been implemented.

The intention of this feature is to advise all interested parties (as defined in escalation profiles for each customer service within the Tracker module) about the impact of a server host outage (database host environments and remote host environments).

When JSM detects that a JSA connection is lost, this not only generates server down alerts (host environments) where applicable but also creates a Tracker fault with a description that a possible server host outage has occurred.

While the server host Tracker fault is open, should the JSA connection be re-established or lost again, this is reflected in updates to the already open call.

Any server down alerts generated due to the JSA connection being lost will identify the appropriate Tracker call reference, if one exists and is still open.

The Tracker fault generated is assigned to the operations role defined in the initialization file, is opened as a priority C and is against the appropriate Hardware Item service component equating to the server host's fully qualified domain name. This Tracker fault may then be used to generate multiple sub-calls for each database environment and remote host environment resident on that server host.

All such sub-calls generated are assigned to the Tracker service corresponding to the host environment and are opened with a priority of A. The intention of this fault explosion is that any escalation profiles matching the open calls will be invoked and will advise all interested parties.

If one doesn't already exist, a server host down Tracker fault can also be invoked manually from the System View screen. If a server host already has a Tracker call associated, then this call can be opened and viewed from the System View screen as long as a Tracker GUI application is running on the same desktop as the JSM GUI.

If a host environment has an exploded fault assigned to it, then this call may also be opened from the System View screen by right clicking on a host environment entity and selecting the Show Host Environment Call #nnn menu option.

Sub-calls may be updated and closed independently from the parent call, however they cannot be reopened.

When a parent call is updated, the user is prompted as to whether the sub-calls should also be updated (regardless of the sub-calls open status). If the parent call is closed, a warning is provided stating that all sub-calls will be updated and marked as closed and an option to cancel the operation. Once a parent call is closed, it may not be reopened.

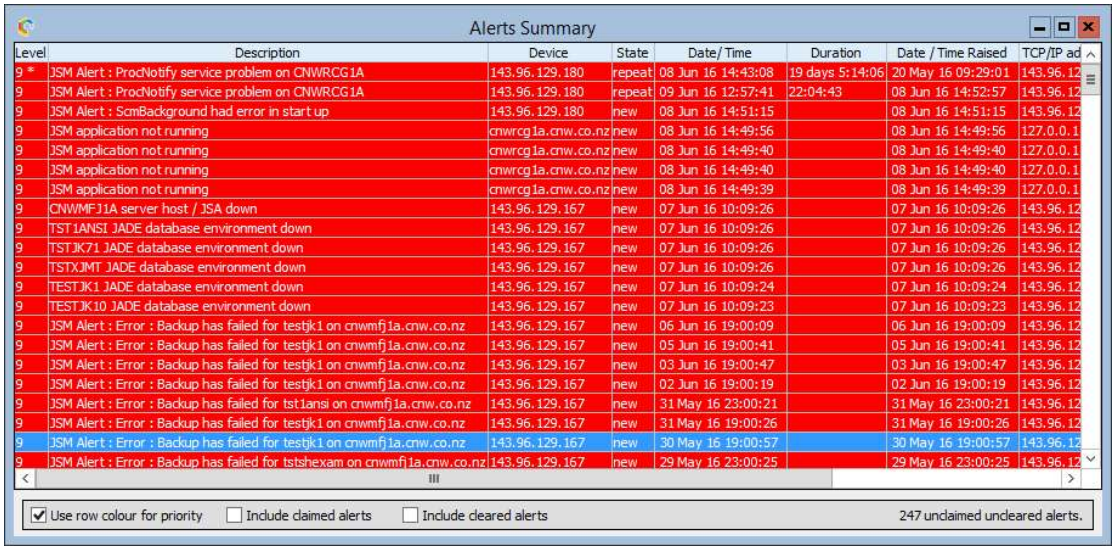
Alert Management

Detail of alert handling available with the following screens is described after the screen shots below. This is generic for all three screens and in the first two is provided as menu options, and in the Alert Details screen as buttons on the form.

Unclaimed Alerts

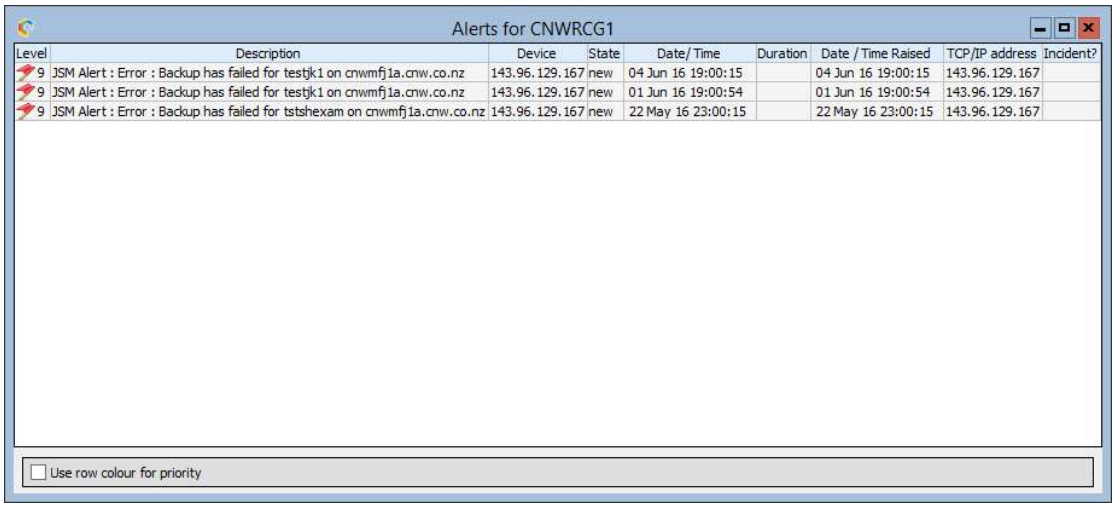
The following screen shot is primarily designed to show unclaimed alerts and is the principal screen for monitoring and handling incoming alerts. The number of uncleared unclaimed alerts is shown in the status line and refreshed on every form update.

Claimed alerts and / or cleared alerts can be added or deleted to the presentation by checking the appropriate selections. The use of colour to differentiate between alert levels as well as active and cleared alerts may also be turned on or off by use of the check box.



Claimed Alerts

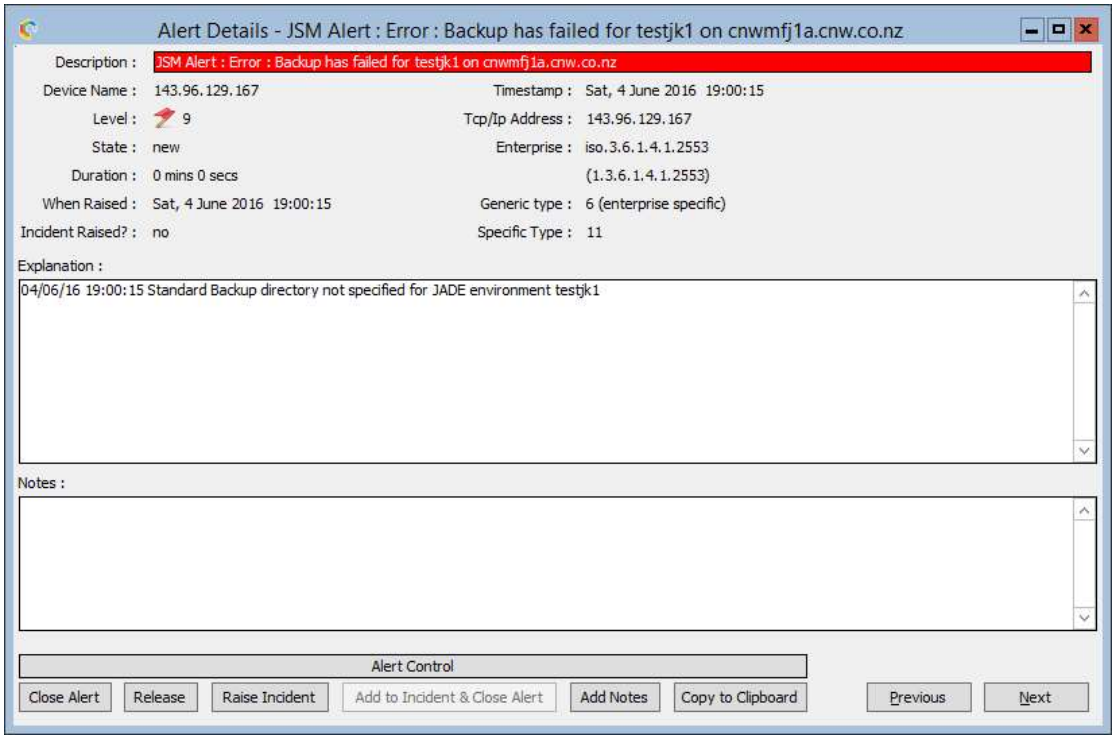
The following screen shows a list of alerts that have been claimed.



Alert Details

Double clicking on any alert in either the unclaimed or claimed alert screen or selecting menu option details (visible as Menu option Actions on MDI Frame or if right click after row selected) will provide the following form showing all pertinent alert information, including any notes that may have been defined.

The alert detail below is an alert that has been claimed as the Claim button is not visible and all the buttons shown are enabled.



Entering Notes

Operators may add notes to describe what actions were taken with an alert. Notes are also added when a JADECare Tracker call is raised (via Raise Fault) or if alert details were added to an existing Tracker call (via Add to Fault and Close).

Alert Handling

Any actions taken with an alert (examining, claiming, closing etc) are recorded as part of the alert history, which can be displayed after the fact.

When viewing any of the above three forms, options are provided to action the alert. These are :

- display alert details
- claimand close the alert
- claim the alert
- close the alert
- release the alert (return it to the unclaimed pool)
- add notes
- raise a fault (manually or via SARMS if implemented and active on desktop)
- add to an existing fault and close alert (only if SARMS implemented)
- copy alert details to clipboard.

Display Alert Details

Refer to Alert Details above.

Claim and Close Alert

This is a new action provided in JSM 3.8.06. This allows an alert or multiple alerts not already closed or cleared to be claimed (even if claimed by someone else) and closed in one action. Refer to details under ‘Claim Alert’ and ‘Close Alert’ for further details.

Claim Alert

Before any further action can be taken, an alert must be claimed. A user defined as a super-user (refer to roles) can take over an alert claimed by someone else. However an alert must be

claimed before it can have further notes added, cause a fault to be raised, have details added to an existing fault etc.

On the Unclaimed Alert screen, a single row or multiple rows may be selected and may be claimed by this user by simply clicking Claim Alert

Add Notes

One or more alert rows can be selected with this option.

The form shown under heading above ‘Entering Notes’ will be presented.

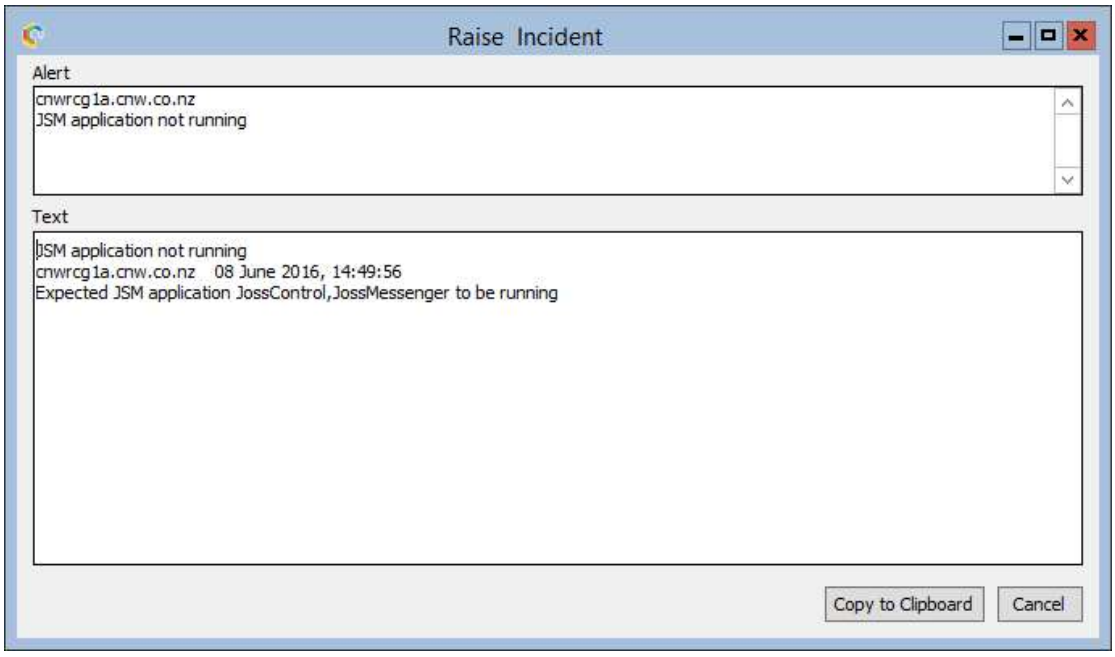
Once the Note form has been OK’ed, all alerts that were selected will have the note details entered added to each of them.

Raise Incident

Note that only alerts that have been claimed and have not had a fault raised against them may use this option.

If JADECare Tracker has been implemented and a thin-client is also active on this desktop, then multiple rows may be selected otherwise only one row may be selected.

If a JADECare Tracker thin-client application is not active on the desktop, then the following form will be presented and the Copy to Clipboard function will provide a clipboard image for pasting into a fault and will mark the alert as having raised a fault.



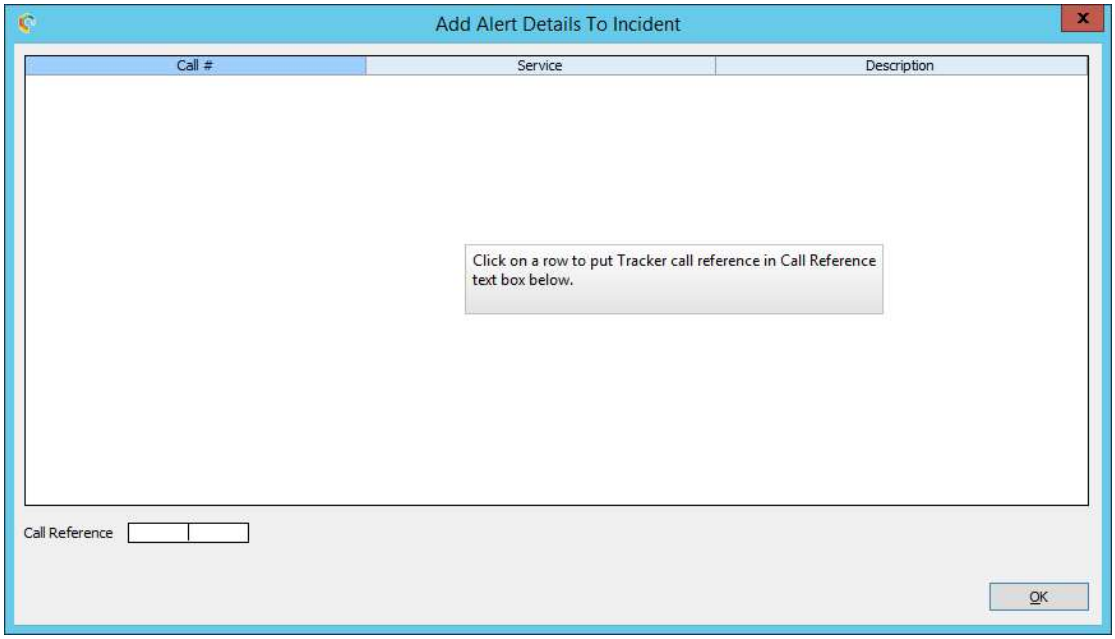
If JADECare Tracker is implemented and a thin-client application is active on this desktop for this user, then the alerts selected will be concatenated together (in when raised order) and cause a SARMS new call screen to pop-up on the desktop. The operator is required to, at minimum, specify the customer, service or component and assignee. When the call is committed, the alerts concerned are updated with notes detailing that a fault has been raised and the call reference and the alerts are left open.

Add to Incident

Note that only alerts that have been claimed and have not had a fault raised against them may use this option.

If JADECare Tracker is not implemented, then this option will be disabled.

One or more rows may be selected and the Tracker call required may be selected from the following form:



The Tracker calls that this user has opened or updated in the current logged on session will be displayed here. A call may be selected by clicking on a table row which will place the call reference in the text box shown and display the call service and call description for verification purposes and enable the OK button.

A call reference may also be keyed in and by tabbing out of the call reference field, the call service and call description of the call will be displayed for verification purposes and the OK button enabled. If the call reference keyed in is not a valid SARMS call, then an error message will be displayed in place of the call description and you will be required to rekey the reference or select from the table.

When the form is OK'ed, all alerts details will be concatenated into another call history element, the assignee concerned will be emailed, the alerts concerned will be updated with notes advising their details were added to fault and the fault number and the alerts will be marked as closed.

Close Alert

One or more claimed alerts may be selected for closure.

If one or more of the alerts has not been cleared and have no notes associated, then the "Notes' form will be provided to enter a closing note.

If all the alerts selected have been cleared, then the alerts will be closed immediately.

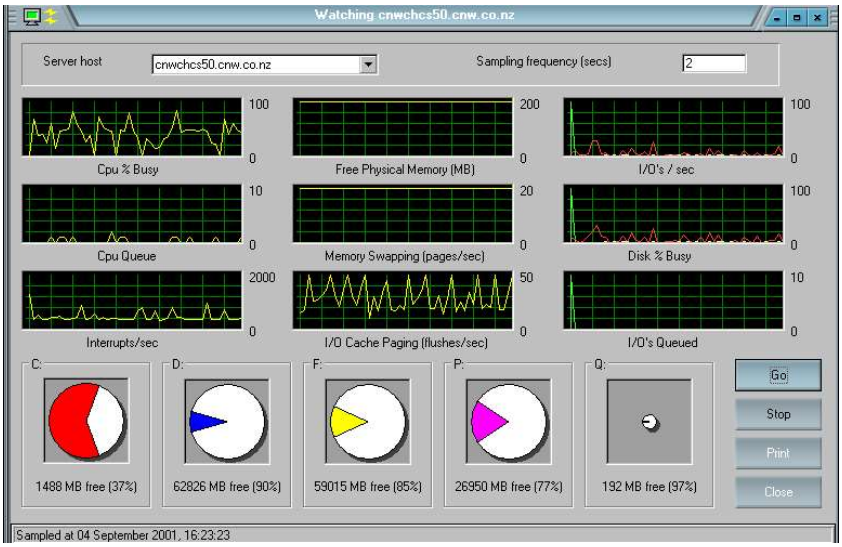
If all the alerts selected have one or more notes associated, the alerts will be closed immediately.

Cleared Alerts

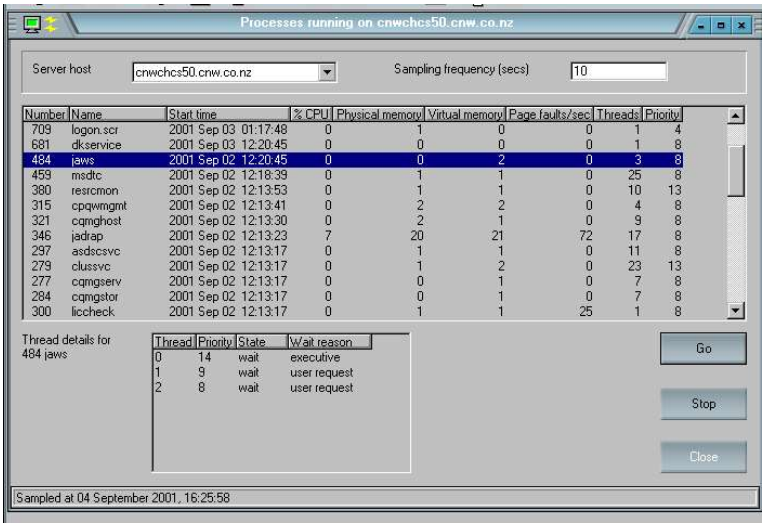
These are alerts that have previously been raised as problem but have since been resolved i.e.: A server down alert has been raised, once the server re-establishes a connection to JSM the server down alert clears automatically.

Cleared alerts up to one day prior to the current date are closed automatically with each scheduled Janitor run.

Real Time Monitor



The above display is sampling the requested host for host wide information at the sampling frequency specified. The graphs provided reflect the items being sampled and the pie charts show the disk utilisation. The colour of the pie charts reflects each disk drive being sampled and these colours are also depicted in the I/O based graphs to show I/O activity for each disk drive.



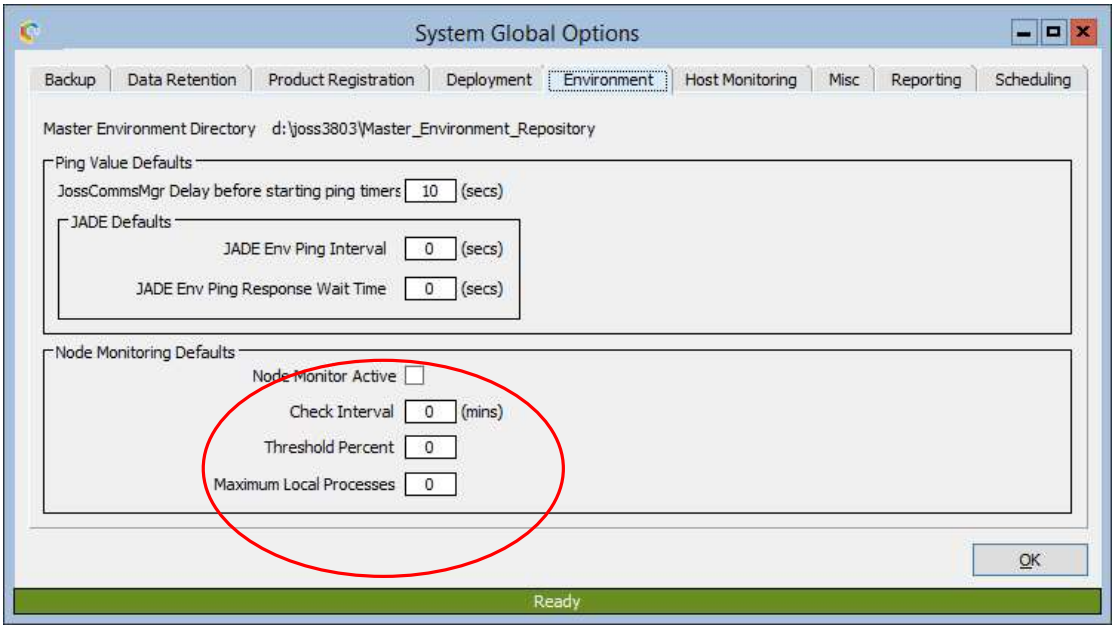
The above screen shows processes running on the host nominated and is refreshed at the sampling frequency specified. Each process can be further expanded on for its thread details by clicking on the required process.

Node Process Monitoring

This feature provides an alert mechanism for when the nominated threshold of JADE processes for a JADE node exceeds the maximum local processes allowed.

Global Settings

The feature may be enabled or disabled globally via the **System Options** form (see below)



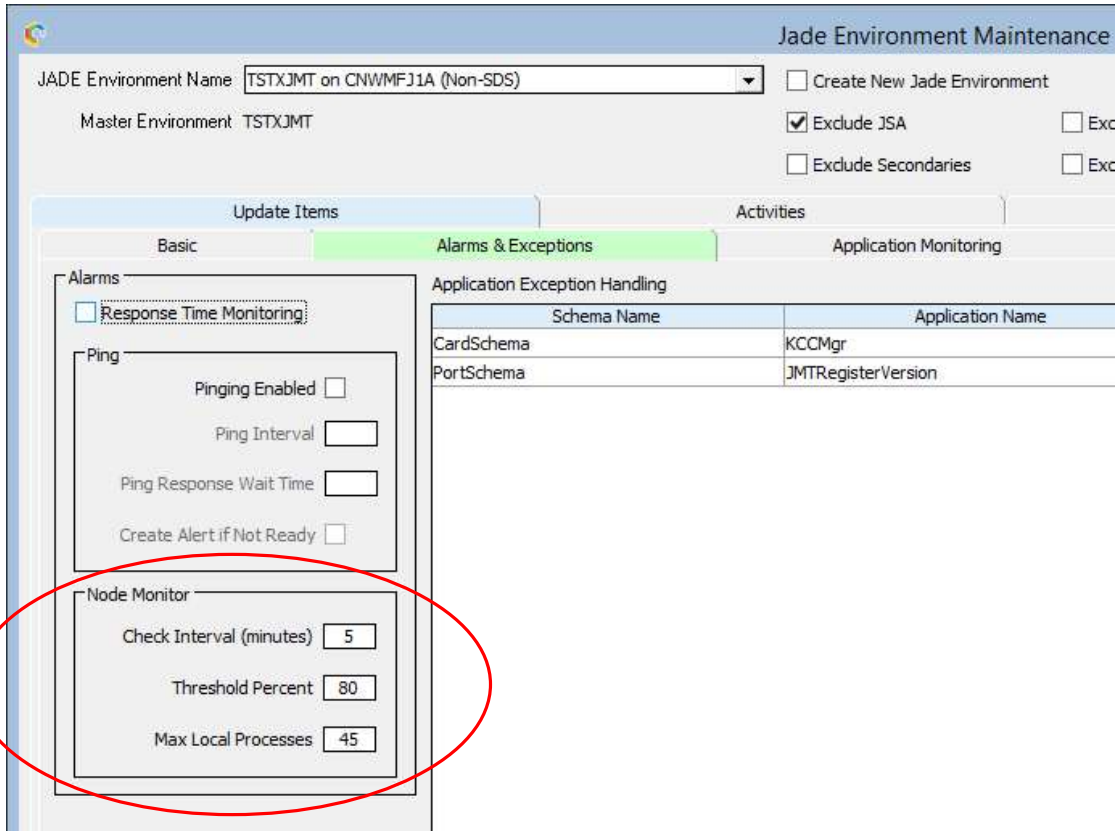
All new host environments created (either by discovery or manually via the Host Environment Maintenance form) inherit the global values provided here.

The values are as follows:

Node Monitor Active	This options will turn the node process monitoring on or off for all host environments. This value is replicated in all JSAs so that they operate autonomously should the JSM connection be lost.
Check Interval	The frequency (in minutes) as to when each node is to be sampled for number of user processes running.
Threshold Percent	The threshold percentage value of each node's maximum local processes which, if exceeded, will cause an alert to be generated.
Maximum Local Processes	This value is used as the default maximum local processes if the node's ini file does not contain an explicit **MaxLocalProcesses** setting

Host Environment Settings

Each host environment will be initialized with the global option values but may be individually defined or turned off.



To turn off node process monitoring for a specific host environment, set any one of the values below to zero.

Check Interval	The frequency (in minutes) as to when each node is to be sampled for number of user processes running.
Threshold Percent	The threshold percentage value of each node's maximum local processes which, if exceeded, will cause an alert to be generated.
Maximum Local Processes	This value is used as the default maximum local processes if the node's ini file does not contain an explicit **MaxLocalProcesses** setting. The ini file value for the specific node is always used if available.

Host Environment Node Monitoring

Every **Check Interval** minutes, a process in the host environment checks the number of user processes per node against the calculated threshold for each node. The calculated threshold is derived as **MaxLocalProcesses * ThresholdPercent div 100**, where the MaxLocalProcesses is either the default value defined in JSM or the MaxLocalProcesses value as defined in the ini file for that node.

Repeat alerts will be issued every **Check Interval** if the node concerned is still exceeding the calculated threshold. If an alert was issued and the next check finds the calculated threshold is no longer exceeded, then the alert will be cleared. The alert contents are as follows:

Short description

```
<env name> : Processes per node threshold exceeded
```

Long description

```
<x> processes detected for node (PID=<pid>) on <node name> (<node
type>), exceeds threshold of <threshold value>
Open a Priority B fault and assign to the CSR
```

When node process monitoring is active for a host environment, a local file is placed into the **JournalRootDirectory** of the environment with title of format NodeMonitor_<env name>_<ccyymmdd>_<hhmmss>.csv. This file is closed at the start of each new day and contains one entry per node for each check interval. Details contained are as follows

Date	Format CCYY/MM/DD
Time	Format hh:mm:ss
PID	The process id of the JADE node
MaxLocalProcesses	this will be either the default value specified for the host environment or the value as obtained by the node from the **MaxLocalProcesses** setting in its ini file
Actual	Indicates which value the max local processes above is: Yes – actual value from ini file <blank> – value is default ex JSM
NodeName	The computer name and unique id assigned by JADE for the node.
Total	Total number of user processes running
Not Registered	Number of user processes that have not registered (ie have not activated Karma Control)
Exceeded	This will contain 'Yes' when the percentage threshold of max local processes has been exceeded otherwise will be blank.

Host Environment and Karma Control

As part of the 'Node Process Monitoring' feature, there is also the ability to identify user applications within a managed Host Environment that do not activate Karma Control. Be default, every user application in a managed environment must activate Karma Control to allow visibility to the application by the JADECare management software.

When a new JADE application starts (which creates a new JADE process), if 60 seconds elapses and the process has not registered in that time frame (by executing app.cnActivateKarmaControl and with ini file setting **[Karma] KarmaControl=true**) then an alert is generated. Subsequent executions of the

same JADE application will cause repeat alerts to be generated identifying the nodes.

The alert format is

Short description

```
<env name> : application has not registered and cannot be managed.
```

Long description

```
<schema name>::<app name> instance <x> from node <node name>
(PID=<pid>) detected as running but has not activated Karma
Control.
Open Priority B fault to development group to rectify.
```

The alerts for specific applications can be bypassed is really required by including an ini file option for each as follows:

```
[Karma]
IgnoreKarmaControlCheck<n>=<schema>,<application>
```

Application Deployment

Overview

The JADECare Systems Manager provides the deployment and installation of changes to JADE Managed Systems. These changes can include any one or more of the following

- schema changes (schema, class or method patches)
- command files (jcf)
- reportwriter changes
- binaries and miscellaneous files
- JADE version, consolidated patches or hot fix upgrades
- CardSchema version upgrade
- Specifically named batch scripts to run at key stages of the deployment

Further, multiple individual deployments to a UAT system can be bundled together and applied in exactly the same order to a production system as part of one overall deployment.

The definition of the changes to be implemented is known as a JADE Authorised Release Instruction (JARI), the content of which is defined via the JARI Wizard and authorised by a person acting on the client's behalf.

WARNING: User supplied patch files with file titles (excluding path) longer than 50 characters will have this file name truncated when viewing the user patch history.

XCF Log Monitoring During Deployment Assembly

With JADECare 24 using the JADECare Systems Manager (JSM) in a SDS Environment, there are several directories used by JSM that need to be replicated to the SDS Secondary's location. One of these is the Master Environment's staging area (see **System Global Options** tab **Deployment** labelled JSM deployment repository). As the deployment assembly process creates and uses files in this staging area, there are some conflicts arising with **Commander** which manages the **XCF** function.

The 3.7.02 JSM release provides the ability to issue requests for **Commander** to stop or start log monitoring the directory concerned. This feature can be enabled or disabled by changing ini file option **[JadeSystemsManager] DisableXCFLogMonitoringDuringDeployment=[true | false]**.

Deployment and Reorganization

SDS Secondary Ini File Options

[Karma] ResumeTrackingWaitInterval

The SDS Secondary deployment periodically checks the tracking status of the target environment and will issue 'resume tracking' requests.

There have been occasions noted where the data pump application has taken a longer than expected time to move past a tracking state after being told to resume and had led to deployments maked as 'in error'.

The defult period in which the deployment engine will wait to request the next tracking state after issuing a resume tracking is as follows:

4.0.02 and later JSA:

- 5 secs – for SDS Secondary Native
- 10 secs – for SDS Secondary Relational

4.0.01 JSA:

- 2 secs – for SDS Secondary Native
- 5 secs – for SDS Secondary Relational

The default values can be overridden a the above ini file setting in the target environment's to tailor this setting.

Min value – 10 secs

Max value - 30 secs

Reorganization Ini File Options

There are 2 known ini file options where the setting can cause the deployment process to fail unconditionally. In the case of an offline deployment, this would require a restore from pre-deploy backup.

The ini file options are:

- [PersistentDb] NoReorgRecovery
- [JadeReorg] FastBuildBTreeCollections

The first option **NoReorgRecovery=true** is invalid for use during an online deployment and should not be used at all for SDS Primary environments.

The second option **FastBuildBTreeCollections=true** is only valid for offline deployments and then only if the **JadeReorgApp** or **jadloadb** commandline explicitly states '**reorgAllowUpdates=false**'.

As from the 3.7.02 release of JSM/JSA, the above options are now controlled by the deployment process (by updating the ini file) depending on:

- The deploy being online or offline
- Whether the schema entity being loaded contains literal _fastbuild (see below).

The ini file option **[PersistentDb] NoReorgRecovery** will ALWAYS be set to false by the deployment engine for all deployment scenarios.

Refer to section **Reorganization option 'FastBuildBTreeCollections'** for more details as to how FastBuildBTreeCollections option can be used.

Reorganization option 'FastBuildBTreeCollections'

The **FastBuildBTreeCollections** type of reorganization may only be used by offline deployments.

The above reorganization option is now triggered by the title of schema entity being loaded, that is when it contains the (case-insensitive) literal '_fastbuild'.

NOTE : Online deployments will ignore this literal within the schema entity title as this reorganization type is invalid for online reorganization.

If a schema entity is loaded that has the literal '_fastbuild' in its title it will:

- Update the ini file such that **[PersistentDb] NoReorgRecovery=false** and **[JadeReorg] FastBuildBTreeCollections=true**
- Specify '**reorgAllowUpdates=false**' as jadloadb and jadereorgapp command line parameter.

If the JadeReorgApp fails due to the Fast Build option being on and the deployment module is able to detect that this was a Fast Build option issue (this feature is available from JADE 6.3.11) then the JadeReorgApp is retried after turning the FastBuildBTreeCollections option to false.

Optional Deployment File Source

The JARI Wizard and JARI Batch Build requires the user to define the directories which contain the files to be deployed. An additional option is provided allowing the user to deploy from a **Central Deployment Repository** visible to the JSM database server. The use of this option can eliminate the need to transfer large deployment files to the developer's workstation simply to be copied back to the JSM server for deployment.

The Auto-Deployment feature, if used, requires any files to be deployed to be located within the appropriate **Central Deployment Repository** directory structure applicable to the master environment.

If specified in the JARI Wizard / JARI Batch Build, files are copied from the central repository first, then client workstation files are copied. If duplicate file names are detected the user is warned and given the opportunity to proceed or cancel the file copy.

If the auto-deployment option is configured, then the sole source of files is from the **Central Deployment Repository**.

At the conclusion of a deployment (installation completed) which includes files from the central repository, JSM removes all files for the deployed environment from the repository.

The root directory for the **Central Deployment Repository** is defined in the JSM ini file and can be viewed on the **SystemOptions** screen, accessible only to JSM super users.

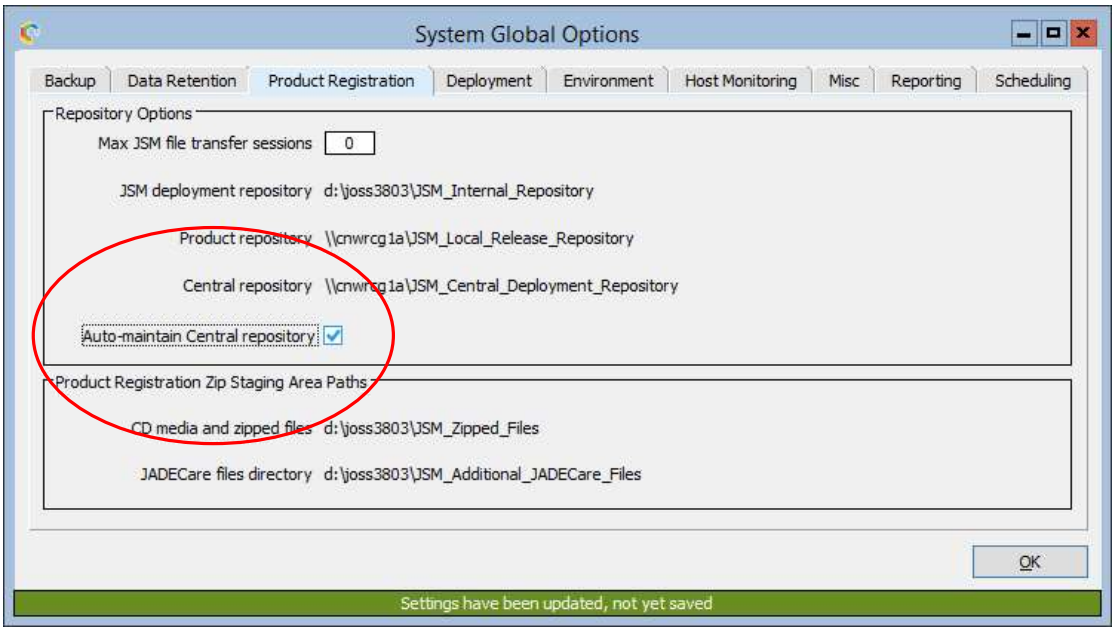
All platform specific directories that are required need to be defined, the same applies when using ANSI or UNICODE.

The directory structure is as follows:

<three character customer id>
\<environment name>
\SCHEMAFILES
\ANSI
\i686-msoft_win32
\server
\bin
\lib
\misc
\system
\i686-suse_sles9
\server
\bin
\lib
\misc
\system
\UNICODE
\i686-msoft_win32
\server
\bin
\lib
\misc
\system

Maintenance of Central Deployment Repository.

The Central Deployment Repository can be automatically maintained by setting the **Auto-maintain Central Repository** checkbox in the **System Global Options** form under the Deployment tab.



Checking this option on will launch a background application **JossSetupCentralDeployRepos** which will ensure that all master environment directory structures required are established. This application will also ensure that all developer access defined within JSM to master environments is granted as permissions with Full Control to the appropriate master environment directories.

From that point on and while this feature remains turned on, the application **JossGeneralEventMgr** will continue to maintain the **Central Deployment Repository** as follows:

- Additional master environments will have their corresponding directory structures created
- As distributions are added to a master environment, the addition sub-directories will be defined
- When a distribution is removed, the master environment sub-directories will also be removed (usually at end of a deployment)
- When a master environment is deleted, all associated sub-directories and contents will be removed
- When a developer access is added or deleted to a master environment / customer (depending on the Explicit Security flag) the developer permissions will also be granted / removed from the appropriate directory structure.

Nested Deployments

This feature is only available in **offline** deployment modes. It is **not** available with the online deployment.

Only one JADE upgrade may be executed by any JARI either via a child release or from the parent JARI.

To modify a JADE licence key, this must be defined on the parent JARI as typically the child release licence key will be applicable to the environment of the child release and not the parent environment.

Regime Definitions

The concept of a regime has been added to retain existing functionality of deployments until the pre-requisites for regime 2 exist and to allow an orderly transition from one to the other.

Regime 0

This regime is no longer supported as from JSM 3.3.

Regime 2

The Master Environment's regime may be set to a minimum of 2. New JADE Master Environments automatically set this now to 2. The JADECare Systems Agent managing all applicable host environments must all be version 3.1 or later and all host environments must be using CardSchema 3.1 or later.

Regime 2 allows for the automation of:

- SDS Secondary deployments (for Relational databases only a Full Extent database will be supported)
- Automatic fat client deployment
- Multiple O/S platforms for a host environment

It also allows for **online** and **offline** deployment, these are described in more detail below.

Regime 3

This deployment regime builds on the facilities provided by regime 2 but provides a greater degree of control with regard to thinclient downloads.

The impact of thinclient downloads has been quite noticeable and with JADE improvements in this area (from JADE 6.2.18 and JADE 6.3.04) plus the changes to the deployment module, we can now identify what, if any, thinclient changes will be incurred.

WARNING: Changing a master environment to regime 3 mandates the following changes:

- that the thinclient details be downloaded before actually changing to regime 3
- that the first deployment at regime 3 is either an initial deployment or a JADE upgrade.

Online vs Offline Deployment.

Both deployment variants require that the host environment is taken offline at some stage during deployment, however the online deployment option minimizes the period that the environment is offline.

Online:

The order of deployment is different than an offline.

All schema type files are loaded one after the other while the system is running and use **loadStyle=latestSchemaVersion** which means that all schemas affected are versioned.

Once all schemas style files have been loaded, the majority of the reorganization is executed, also while the system is online.

At a pre-determined (scheduled) time, the system is taken down for transition to occur. In many cases this will just mean moving the schema changes from latest version to current version, but may also involve the last phase of a reorganization where collections have been created, modified or deleted.

While the system is down, any binaries etc are put in-place and the system brought back online

Jade release upgrades and nested JARIs can **not** be utilized in an online deployment.

Offline:

Before any deployment changes take place, the system is taken offline.

All binaries and other miscellaneous file are put in place (from Jade upgrade, CardSchema upgrade, user provided).

If Jade upgrade requested, then the Jade upgrade script is executed.

If CardSchema upgrade is requested, CardSchema schemas are loaded with **loadStyle=onlyStructuralVersioning**. If a reorganization is required, then this is performed immediately.

All user schema style files are then loaded using **loadStyle=onlyStructuralVersioning**, and if a reorganization is required as each schema file is loaded, then this is performed immediately before starting the next file load.

Items to be aware of:

As the reorganization for an online deployment is done as a result of loading multiple schemas, the order of classes to be reorganized is defined by Jade and it is potentially possible for data errors to make the reorganization fail, which may not be the case if the deploy were an offline deployment.

Be aware that if the deployment requires certain binaries or other files to be in-place before schema loads etc, then an offline deployment is required.

If a sub-classed control is modified, this requires that a reorganization occur before the database definition file (ddb) can be loaded (jadloadb returns exception 8525 Reorganisation is required prior to DDB load) and can only be performed using an offline deployment.

LoadStyle

Online Deployment

The default **loadStyle** used for loading schemas in online deployments before transition occurs is **latestSchemaVersion**.

Some files may be deferred until transition has occurred, these may be ddb files (when a sub-classed control change means these need to be deferred) and report writer files (which require the current schema view). These deferred files, if schema type files, are loaded with **loadStyle=currentSchemaVersion**.

Offline Deployment - SDS Primary Database

For a SDS Primary offline deployment the default **loadStyle** is **latestSchemaVersion** with the following exceptions:

- load file is a report writer file – **loadStyle=currentSchemaVersion**
- load file is a command file and contains one or more commands that requires **loadStyle=currentSchemaVersion**. These commands are currently:
    - moveinstances
    - rename class
    - create dbfile
    - delete dbfile
    - modify class
    - modify schema
    - delete instances
    - remap class
    - rename interface

The **loadStyle=latestSchemaVersion** requirement is to avoid changes to sub-classed controls causing a 1284 exception on the SDS Secondary and requiring manual intervention to get the SDS Secondary journal replay to continue.

Offline Deployment - non-SDS Database

For a non-SDS offline deployment the default **loadStyle** is **onlyStructuralVersioning** with the following exceptions:

- load file is a report writer file – **loadStyle=currentSchemaVersion**
- load file is a command file and contains one or more commands that requires **loadStyle=currentSchemaVersion**

JADE Command Files

General

When building a JARI, if a schema file provided makes reference to a commands file (type .jcf) then the JARI wizard expects a command file matching the

corresponding schema file name to be provided (note that this may differ from the name given in the schema file). It is the developer's responsibility to resolve any issues with this (e.g. the developer may have to remove line 1 from your schema file if the command file to be required), and to specify when a command file should be loaded (i.e. the order sequence in relation to other schema files).

Command files may be submitted:

- As standalone entity
- With a schema file (the schema file references the command file making it a mandatory file)
- With a class file (the class file references the command file making it a mandatory file)

Regardless of the command files associations, the user can define the load order of command files as for any other file to be loaded into a JADE environment.

Any fatal errors (exit code = 8515) detected while executing the JCF command file will return an exit code causing the deployment to fail regardless of the setting of the AbortOnError commands.

Any processing errors (exit code 8514) where one or more commands did not complete successfully will only cause the deployment to fail if the **AbortOnError** command is not specified or is defined as True.

LoadStyle

There are command variants which dictate that the **currentSchemaVersion** loadstyle be defined. These are:

- Create DbFile
- Delete DbFile
- Delete Instances
- Modify Class
- Modify Schema
- Delete Schema
- MoveInstances
- Remap Class
- Rename Class
- Rename Interface

The **currentSchemaVersion** loadstyle is incompatible with online deployment. A mix of commands using the above and those commands which require **onlyStructuralVersioning** (at minimum) or **latestSchemaVersion** loadstyles is also incompatible.

Such inconsistencies are checked and generate errors at the time the command file is submitted when building a JARI.

The **currentSchemaVersion** loadstyle is used when any one of the above commands are encountered.

Version Control

JARI Wizard

Start Sheet

On this sheet, the environment is selected and short description entered. A longer description may optionally be entered.

Deployment Type – Normal

Only changes required to be sent are deployed, regardless of whether multi OS has been selected.

Deployment Type – Initial Deployment

Required when first deploying to a multi OS platform and is used to ensure that a full consistency for every platform type for the current JADE version is deployed to the database server.

Release Source – Create New Release:

The JARI has no child JARIs selected, all changes are contained within this single JARI.

Release Source – Use Existing Release from Environment

This allows another environment to be selected and for one or more JARIs from that environment to be listed in a table. One or more of these JARIs can be selected creating what is known as a nested JARI (the master JARI and child JARIs being those selected from this table). The JARIs so selected may contain or not any or all of the following:
- a JADE upgrade
- JADE hot fixes
- CardSchema upgrade
- CardSchema hot fixes
- schema, ddfile, methods, classes or command files
- selected files and / or scripts for various directories

See initialisation file for options that affect this area:
[JadeSystemsManager] AllowCrossCustomerDeployment
[JadeSystemsManager] AllowJariSelectNotCompleted

NOTE: Licence details on selected JARIs are ignored. Only licence details on the master JARI, if provided, will be applied.

Each child JARI will all be applied in child JARI sequence and followed by the master JARI request.

The selection of JARIs to be replayed does not preclude the master JARI from defining schema type files to be loaded, JADE upgrade, JADE hotfixes or CardSchema upgrades.

Deployment Options Sheet

The person authorising the deployment may also provide scheduling parameters. Note that these may be overridden by operations via the deployment monitoring form.

When entering licence key details, these values are now validated before the user may proceed further. Should an error exist, an error dialog is shown and this sheet is re-displayed.

Expect thinclient download

This checkbox is only enabled when the master environment is at deployment regime 3.

If thinclient download files are detected and this option is not specified, then the assembly phase will fail.

If thinclient download files are not detected and this option is specified, then the assembly phase will fail.

Check for methods in error after last JADLOAD.
Checking this box will force a check that there are NO methods left in error (and will force those in error to be recompiled before re-doing the check). See 'Methods in Error advice'.

This option may be overridden if the target JADE Environment ini file has **[Karma] IgnoreInstallationMethodsInError=true** defined.

Last Schema Loaded
If the last schema loaded has methods left in error, then operations is advised. See 'Methods in Error advise'.

Methods in Error advice
If 'Check for methods in error after last JADLOAD' identifies methods in error or the last schema in the sequence to be loaded reports methods in error, then operations are provided an option to IGNORE or CANCEL. The IGNORE option allows the deployment to continue. The CANCEL option causes the deployment to go into error state and requires a restore of pre-deployment backup.
The IGNORE option will only be used when :
- deploy marked as critical and developer contacted and developer advises to use IGNORE
- deploy special instructions to operations clearly specify when the IGNORE option is to be used.

Assembly

- May be started immediately
- May be scheduled

Installation

The options are dependent on whether online or offline backup is selected.

- Offline Backup
  - Shutdown system for offline backup and installation at scheduled time.
- Online Backup
  - The online backup start time can be scheduled after which any of the following may occur
  - Immediately shut systems down and install deployed files
  - Wait for operator authorisation before proceeding to system shutdown and installation
  - Wait for specified installation time before proceeding to system shutdown and installation
- If 'deploy_hold.htm' is present in environment 'c_misc' folder, the deploy will hold until that file is removed.

- If offline deploy/installation requested with online pre-deploy backup, installation will wait before system shutdown (after online backup is completed).
- If offline deploy/installation requested but environment is down at the time of deployment, system will wait before pre-install backup.
- If online deploy/installation requested and environment is up and running at the time of deployment, installation will wait before system shutdown (after online backup is completed).
- If online deploy/installation requested but environment is down at the time of deployment, deploy will stop at 'Check System' as it is not allowed.

'deployin.txt' Usage

A user may wish to pre-determine the load order of supplied schemas, classes or methods. An example of this is when a particular method patch references attributes or references on a class that is also being loaded. In this particular case it would desirable for the class to be loaded before the method, using deployin.txt provides this functionality.

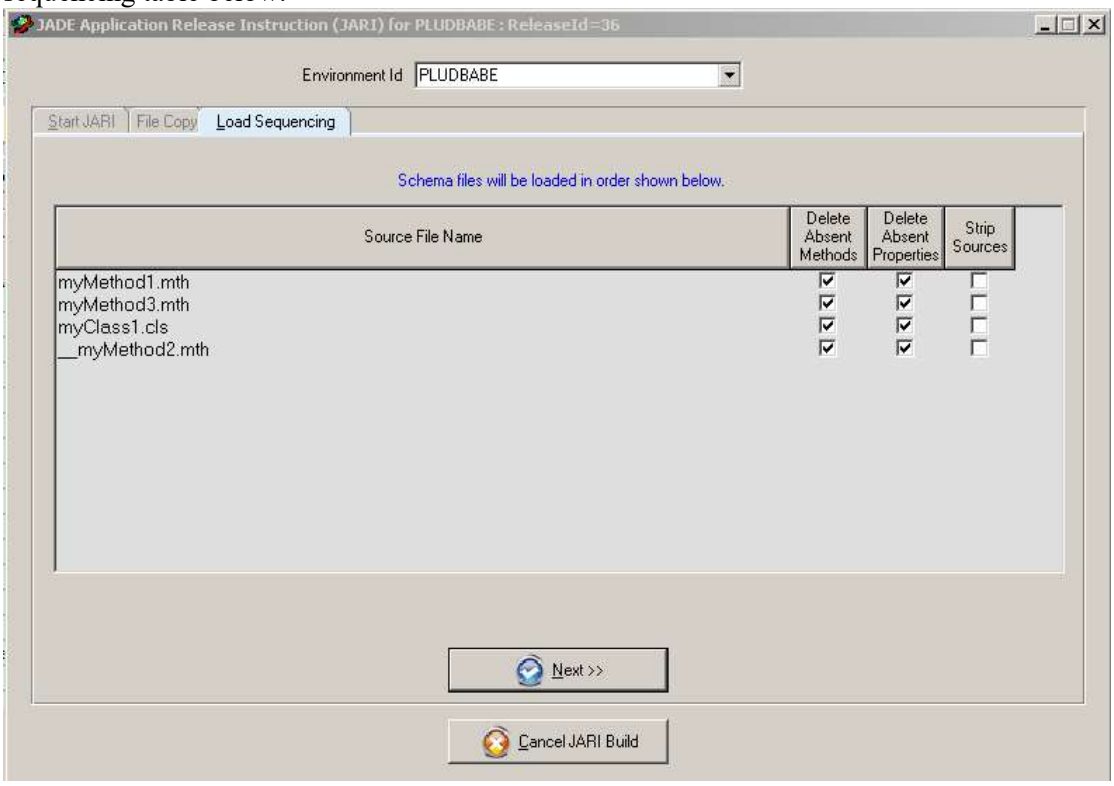
Another reason someone may wish to use a deploy.txt file is in the case of wanting to deploy a large collection of files to multiple systems. For example, 4 test environments that are targeted with the same deployment files. If many files are being deployed then it can be an easy mistake to miss a load submission. Using the deployin.txt file would highlight such an omission during the JARI build stage.

The format of a deployin.txt file may look like this:

```
myMethod1.mth
myMethod3.mth
myClass1.cls
__myMethod2.mth
```

In this example, even though '__myMethod2.mth' comes alphabetically before the other definitions, it will be loaded last. This can be seen in the 'JARI Wizard' load

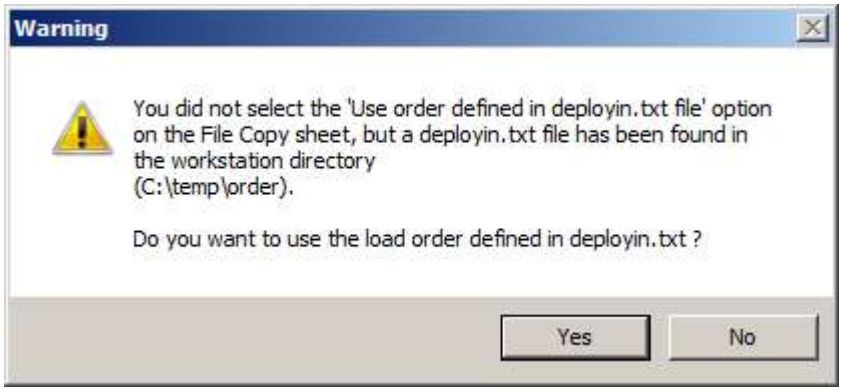
sequencing table below:



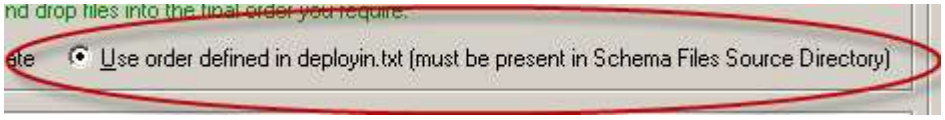
This load sequence can still be adjusted at this point by dragging and dropping the various entries into the desired order.

JSM will check for a deployin.txt file under the following circumstances:

- If a deployin.txt file resides in the same folder as the user supplied class, schema or method source directory and they haven't explicitly asked to use a deployin.txt file they will receive a prompt:



- If a user has specifically requested to use the deployin.txt file via the second sheet on the JARI Wizard:



The deployin.txt file is validated. Declared definitions in the deployin.txt are validated against the user supplied load files.

- If an entry is declared in deployin.txt that has not been supplied as a physical file, an error will occur.
- If a load file is found in the same folder as the deployin.txt and this file is not declared inside the deployin.txt, an error will occur.
- Entries defined in the deployin.txt file are case sensitive, if the definition case doesn't match the physical file case then an error will occur. This needs to be correct for case sensitive file systems i.e. Linux

JARI Batch Build

Overview

The JARI Wizard and JARI Batch Build processes use much of the same logic however, the JARI Batch Build does not issue warnings or permit the user to cancel the JARI build process. The first error encountered is treated as a failure.

The JARI Batch Build process is initiated via the standard JSM GUI interface using the menu items Deployment | Jari Batch Build.

This causes a form to be displayed (see above) allowing the user to specify a directory location.

This directory location can specify the location of where to create a template file or to start the JARI batch build process.

The template file generated (titled example_jaribuild.jri) may be used as the basis for batch JARIs.

The specified directory may contain one or more JARI Batch Build input files, these being suffixed with **jri**. Each of these files is to contain one JARI specification (see Input File below for details) and at completion of each, the file is moved into a sub-directory of the directory specified titled **/success** if JARI is successfully submitted or **/failure** if the JARI specification had an error. If either of these sub-directories does not exist, they will be created as required.

As the JARI Batch Build process runs, it logs all details into a log file jaribuild_<userid>_<ccyymmddhhMMss>.log in directory specified by CardLog LogPath section or JadeLog LogDirectory within the JSM database initialization file.

If the user is connected by thin-client, then a copy of the jaribuild log details are placed in a sub-directory **/logs** (titled jaribuild_<ccyymmddhhMMss>.log).

In order for a JARI to be built via this mechanism, the master environment must define 'Allow JARI Build" to be true in the Master Environment Maintenance form.

As the JARI batch build progresses, the **jri** files are moved into sub-directories **/success** or **/failure** as appropriate, each file being timestamped as it is moved to avoid overwriting any existing files.

Defaults Used

All schema, class and method files loaded via this mechanism default to the following load options:

- Absent methods are deleted (schema and class)
- Absent properties are deleted (schema and class)
- Source code is not stripped

If file **deployin.txt** is present, then the load order defaults to the order defined in that file otherwise it defaults to the order presented by the file system.

Input File

The file is defined in ini file format and is accessed as an ini file. Any misspelled section or verb names will not be detected or cause an error unless the item is mandatory. Comments and items not specifically declared below are ignored.

Keyword	Value	Comment
[ReleaseControl]		Section name
Environment=	<env name>	Master environment must exist **Required item.**
ShortDescription=	<one line description>	Will be truncated to 50 bytes if longer **Required item.**
LongDescription1=	<first line>	
LongDesciption2=	<second line concatenated with first separated by space>	
LongDescription<nn>=	<subsequent lines concatenated with earlier lines>	
InstallationMode=	<Offline/Online>	default = Offline
UseExistingReleaseEnvironment=	<env name>	Master environment must exist Not permitted if Installation Mode is Online.

Keyword	Value	Comment
UseExistingReleaseId1=	<nn>	Release must exist
UseExistingReleaseId2=	<nn>	
UseExistingReleaseId<nn>=	<nn>	
[DeployFiles]		Section name
SchemaFilesDir=	<directory of schema files>	may include deployin.txt to define names and load order, otherwise defaults to file name order.
<platform>ServerBin=	<directory for this platform of binaries to be deployed to database server>	Platform must be valid and registered in JSM and be valid for environments defined
<platform>ServerLib=	<directory for this platform of libraries to be deployed to database server>	Platform must be valid and registered in JSM and be valid for environments defined
<platform>ServerMisc=	<directory for this platform of misc to be deployed to database server>	May include deployment batch scripts
<platform>ClientBin=	<directory for this platform of binaries to be deployed to fat client(s)>	
<platform>ClientLib=	<directory for this platform of libraries to be deployed to fat client(s)>	
<platform>ClientMisc=	<directory for this platform of misc files to be	

Keyword	Value	Comment
	deployed to fat client(s)>	
<platform>ThinClientBin=	<directory for this platform of binaries to be deployed to thin clients>	
<platform>ThinClientLib=	<directory for this platform of libaries to be deployed to thin clients>	
<platform>ThinClientMisc=	<directory for this platform of misc files to be deployed to thin clients>	
[JadeSoftwareUpdates]		Section name
JadeRelease=	<n.n.nn>	Must be registered release and greater or equal to current JADE version. This identifies the release to upgrade to and from which hot fixes are to be obtained.
JadeHotFix1=	<nn>	If JadeRelease is not specified than the release defaults to the current JADE Release. Must be valid hot fix for applicable Jade Release.
JadeHotFix2=	<nn>	
JadeHotFix<nn>=	<nn>	
[CardSchemaUpdate]		Section name
CardSchemaRelease=	<n.n.nn>	Must be a registered release and greater or equal to current CardSchema version

Keyword	Value	Comment
CardSchemaHotFix1=	<nn>	If CardSchemaRelease is not specified than the release defaults to the current CardSchema Release. Must be valid hot fix for applicable CardSchema Release.
CardSchemaHotFix2=	<nn>	
CardSchemaHotFix<nn>=	<nn>	
[Notifications]		Section name
DeveloperId=	<operator userid>	Must be a valid operator **Required item.**
WorkPhone=	<nnn>	default = work phone from developer id
Extension=	<nnn>	default = extension from developer id
CellPhone=	<nnn>	default = cell phone from developer id
PagerService=	<nnn>	default = pager service from developer id
Pager=	<nnn>	default = pager from developer id
HomePhone=	<nnn>	default = home phone from developer id
EmailAddress1=	<nnn>	default = email address from developer id
EmailAddress2=	<nnn>	
EmailAddress3=	<nnn>	
SpecialInstructions1=	<first line>	
SpecialInstructions2=	<second line concatenated with first separated by space>	

Keyword	Value	Comment
SpecialInstructions<nn>=	<subsequent lines concatenated with earlier lines>	
[DeploymentOptions]		Section name
ProductionMode=	<true/false>	default = false
CertifyDatabaseBeforeLoad=	<true/false>	default = false
CertifyDatabaseAfterLoad=	<true/false>	default = false
CheckMethodsInErrorAfterLastLoad=	<true/false>	default = **true**
ReorgExpected=	<true/false>	default = false
AutomaticSystemRestart=	<true/false>	default = **true**
SaveJariForRepeatDeployment=	<true/false>	default = false
ManualBackupRequired	<true/false>	default = derives default from master environment flag (see Manual Backup document)
DefeatPreInstallationBackup=	<true/false>	default = false, if master env production flag on this option will be ignored If ManualBackupRequired then regardless of setting, this is changed to true
DefeatPostInstallationBackup=	<true/false>	default = false, if master env production flag on this option will be ignored If ManualBackupRequired then regardless of setting, this is changed to true
ThinClientDownloadExpected=	<true/false>	default = false, if master env deployment regime = 2

Keyword	Value	Comment
		this option will be ignored
PrimaryLicenceName=	<licence name>	
PrimaryLicenceKey=	<licence key>	
SecondaryLicenceName=	<licence name>	
SecondaryLicenceKey=	<licence key>	

Progress Form

During the JARI Batch Build cycle, the following progress form is displayed and updated.

Automatic Deployment

Overview

A requirement can exist where a development group wants to perform daily deployments to test environments that contain the latest code changes for that day. The auto-deployment feature allows the developer to extract their schemas, miscellaneous and binary files into a predetermined directory structure and setup a schedule within JSM so these files are:

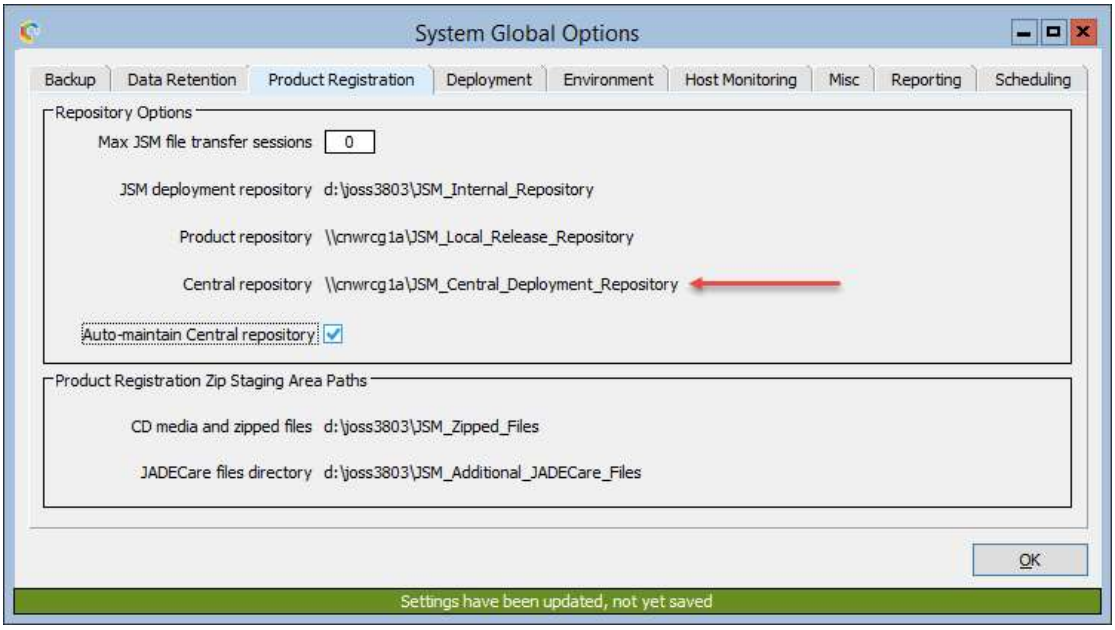
- automatically discovered and assembled into a JARI
- automatically and immediately deployed once the JARI has assembled
- advised of any failure or success via email

Caveats

- cannot be used for Production environments (unless ini file option [JadeSystemsManager] AutoDeployProductionOverride=true)
- deployments are offline (backups are online)
- deployment defaults are always used
- the assembly, distribution and installation steps follow each other with no scheduling. The use of '**system busy**' sentinel and / or '**deploy hold**' sentinel files and / or '**autodeploy_hold.htm**' sentinel can be applied if required.

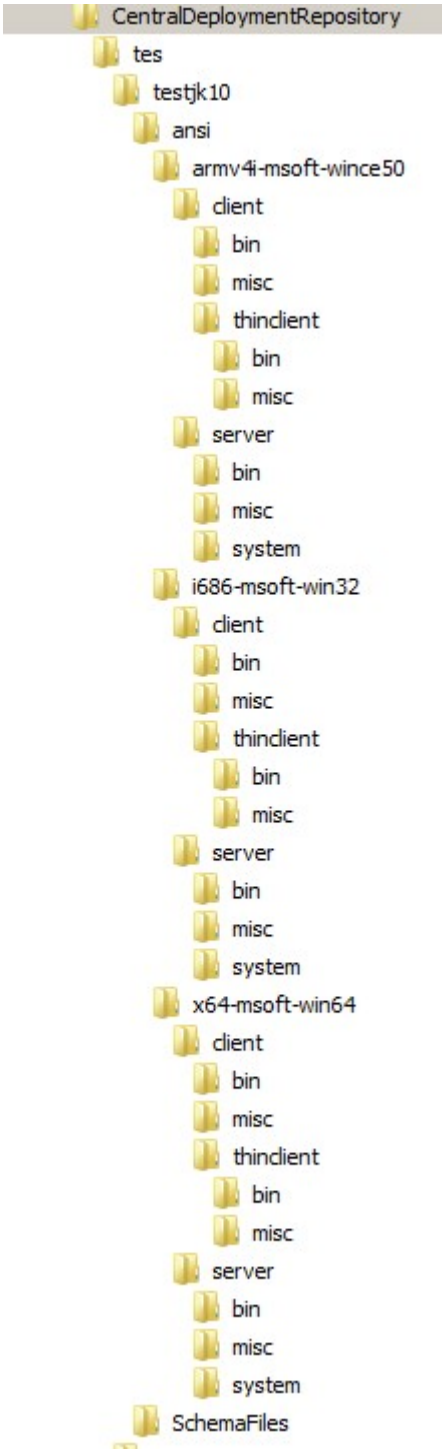
Setup

The auto-deployment feature searches for deployment files in the 'Central Deployment Repository'. The base of this directory can be observed from the 'System Global Options' form within JSM **Options > System Global Options**.



If you don't have visibility to this form you may need to contact your operations staff to advise you of the directory location.

Under the Central Deployment Repository should be a matching customer id for the environment you intend to deploy to. For example a common directory structure may look like :



Customer is 'TES'. Target deployment master environment is 'TESTJK10'

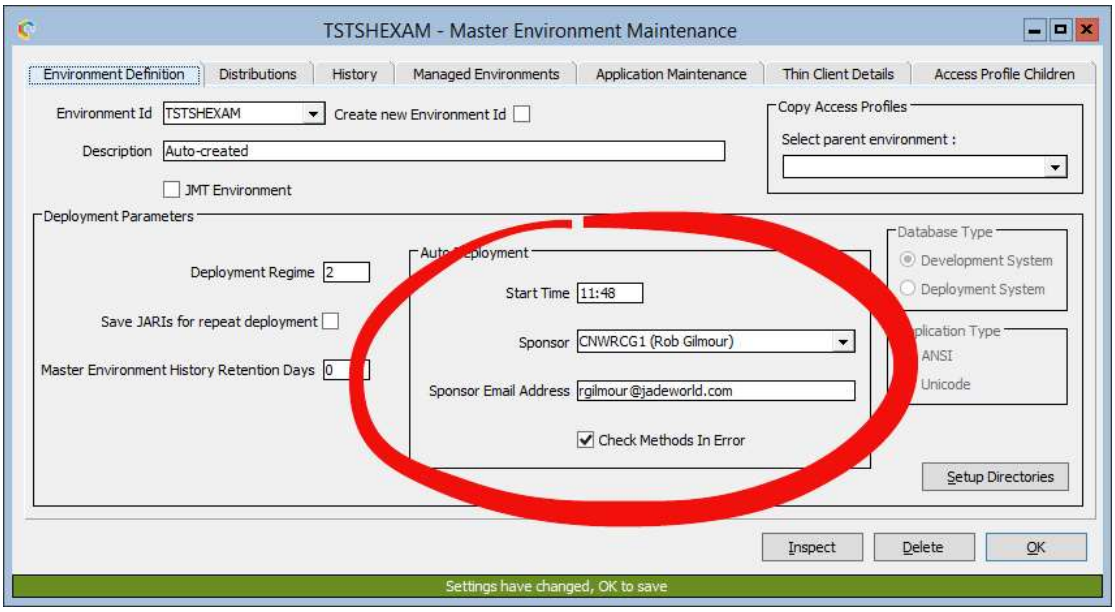
Under this we have 'SchemaFiles' and 'ansi' as the master environment for TESTJK10 is defined as being an ANSI implementation rather than Unicode. Under 'ansi' are all the distribution directories that are applicable to our target environment.

A developer would place their .scm, .ddb, .cls, .mth, deployin.txt and user batch scripts under the 'SchemaFiles' location. User binary and miscellanous files destined for the database server go in the respective **server/bin** and **server/misc** location for the distributions that are required. Thinclient user binary and miscellaneous files go into the **client/thinclient/bin** and **client/thinclient/misc** locations for the applicable client distribution.

The use of **client/bin** and **client/misc** is a hold-over from earlier JADE deployment strategies and was used to define binaries and misc files specifically for remote fat-clients. No files need be placed in these directories.

Scheduling

To schedule an auto-deployment bring up the 'Master Environment Maintanence' form **Setup > Master Environments**.

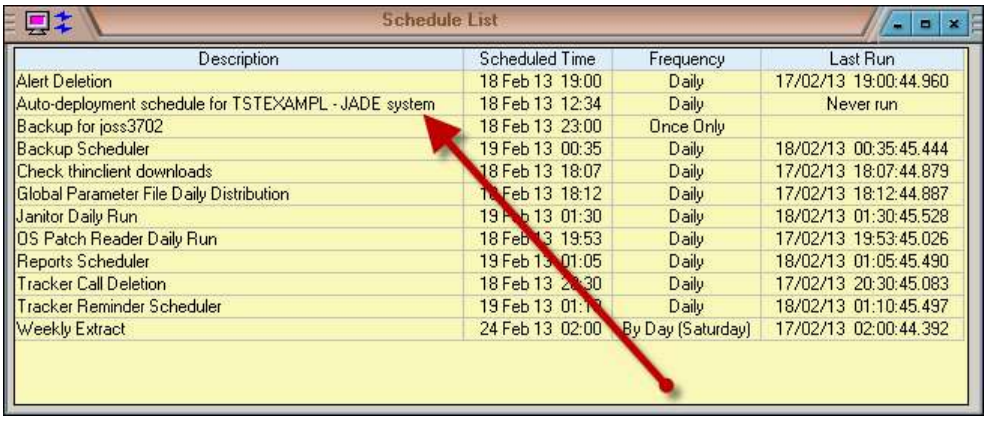


From here you can establish an automatic deployment schedule. The auto deployment feature requires 3 components :

Start time (24 hour)	When the auto-deployment job will initiate every day.
Deployment sponsor	The operator id designated as the deployment overview. Typically this would be the person responsible for oversight on this auto-deployment process. The combo box is pre-populated with: - all master customer staff defined with security roles of Superuser or Operator - any staff with access to the environment and with security roles of Developer, CSR or IT Manager
Sponsor email address	By default this will be the email address associated to the deployment sponsors operator id. However, this email address can be overwritten if desired. In addition further email addresses may be added if separated by a semi-colon ';' This email address(s) will be used for all deployment updates, be it success or failures.
Check Methods In Error	This option is specific to the auto-deployment process only. If true, then after the last schema is loaded then all schemas are check to identify any methods in error, and, if so, the deploy suspends requiring operator intervention to accept or to abort. If the option is false then no check is made for methods in error.

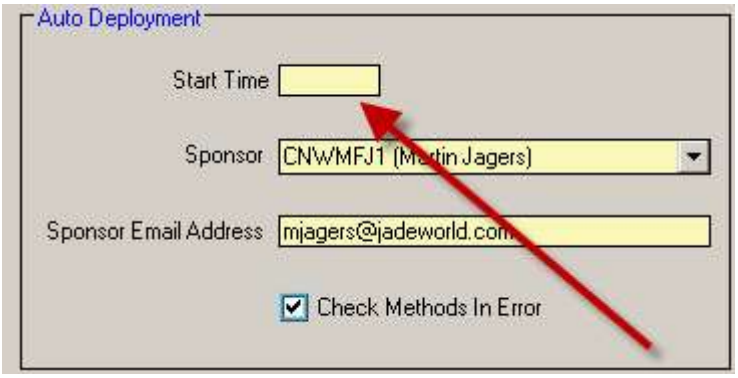
NOTE : This option may be overridden by ini file option [Karma] IgnoreInstallationMethodsInError=true.

Clicking the 'OK' button commits the auto-deployment schedule. Deployment schedules can easily be viewed by bringing up JSMs 'Schedule' form **View > Schedule**.



To change a already established auto-deployment schedule simply update the start-time. This change in time will automatically be reflected in the 'Schedule List' shown above.

To cancel an existing schedule simply remove the start time :



The 'Sponsor' and 'Sponsor Email Address' and 'Check Methods in Error' will remain. This is intended behaviour.

Runtime Checks

When the scheduled time arrives some preliminary checks are made.

Deployment files exist?	If no files exist under the 'Central Deployment Repository' the auto-deployment process logs the fact that nothing was deployed, terminates normally and is re-scheduled for the next day.
Sentinel file exists?	It's possible to halt the auto-deployment job by placing a '**autodeploy_hold.htm**' sentinel file in the 'Central Deployment Repository' directory under the environment name. In the examples shown above this would be the 'PLUDBABE' directory.

	If this file exists the auto-deploy process will terminate normally and is re-scheduled for the next day.
Existing JARIs?	If JARIs exist for this environment in a state other than 'Cancelled' or 'Completed' then the auto-deploy process will terminate and the deployment sponsor will be emailed with a failure notification. It is up to the deployment sponsor to cleanup any existing JARIs in order for the auto-deployment to proceed normally the next day.

Once the JARI is established the usual deployment process checks come into play, if a backup or compaction are running for example this will cause a failure in the deployment process.

Cleanup

When developer supplied files in the 'Central Deployment Repository' have been successfully built into a JARI these files will then be removed from the 'Central Deployment Repository' automatically. This is to clear the way for future deployment files. Should the remove of these files fail, an email of these details will be sent to the deployment sponsor email address(s).

JMT Deployment

This deployment methodology is only available for JSM implementations in the JADECare 24. The menu item will not be visible to other than JADECare 24 personnel.

This deployment option is only available for environments that are explicitly defined as being a 'JMT Environment' (see Environment Maintenance form).

The JMT Environment combo box is loaded with all those environment's that are defined as being a JMT Environment.

When an existing environment is selected, the application release is derived from the first database associated with the environment. This allows the derivation of which target releases are allowed, which populates the Target Release combo box.

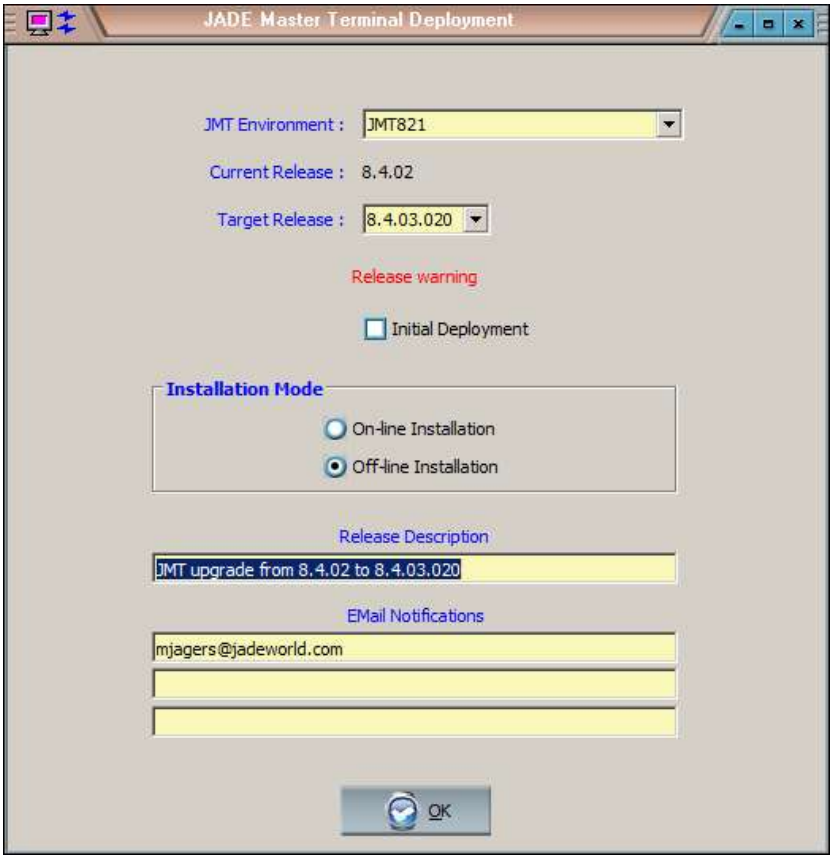
When one of the releases in the Target Release combo box is selected, the 'Installation Mode' and 'Release Description' are derived.

The following rules apply to the derivation of 'Installation Mode':

- current release is a base release and target release is a consolidated release for the same base release – ONLINE
- current release is a consolidated release and the target release is also a consolidated release where both are for the same base release – ONLINE
- all other options mean the 'Installation Mode' is OFFLINE

This does not prevent the user selecting a different 'Installation Mode'.

Setting 'Initial Deployment' means that all of the files registered for the target release as well as the current JADE Release and CardSchema release will be deployed. No attempt will be made to rationalize the deployment package.



The 'Release Description' and 'Email Notifications' can be replaced as required.

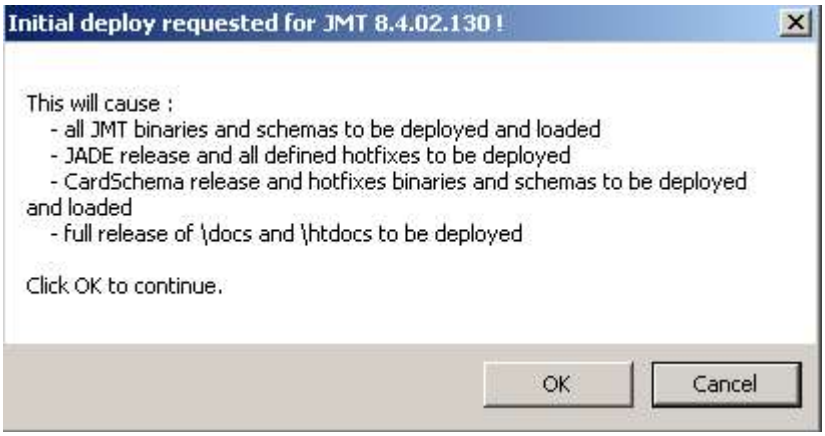
Clicking OK is all that is required to create a JARI for the selected Environment.

This methodology will (if 'Initial Deployment' not set) will try to rationalize the deployment package sent such that only the required changes/files are issued to the JADE Environment(s).

JMT 'Initial Deployment'

When 'Initial Deployment' is selected then a warning dialog is issued advising what will occur should the deployment request be confirmed.

Example:



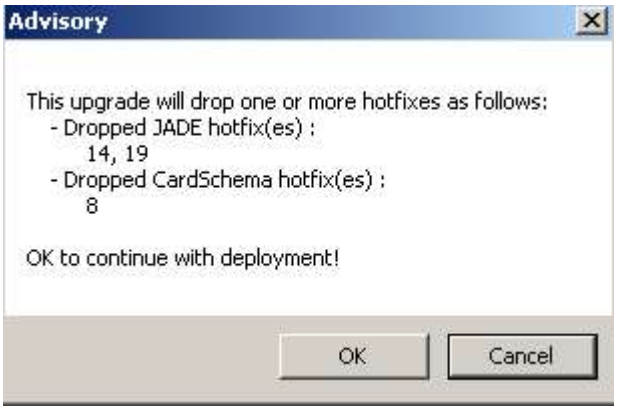
JMT Warning 'Hot Fixes Dropped'

When upgrading from one JMT release to another (ie base release to consolidated release or consolidated release to consolidated release), it is possible for one or

more JADE and/or CardSchema hot fixes to be ‘dropped’. This occurs where the JADE / CardSchema versions are the same and the current release has one or more hot-fixes defined that are not defined in the target release.

Where such an event is detected a warning dialog is issued which requires confirmation to continue.

Example:



JARI List Form

The 'JARI List' form **Deployment > List JARIs** contains an additional column to show what process was used to initiate the JARI. It now looks like this and is informational only :

JARIs established prior to JSM 3.7 will show 'Pre-JSM 3.7'. JARIs initiated from a batch script will show 'Script'. JARIs initiated from the 'JARI Batch Build' process will show 'Batch Builder'.

Authorizing and Scheduling Deployments

There are now two methods for JARI approvals:

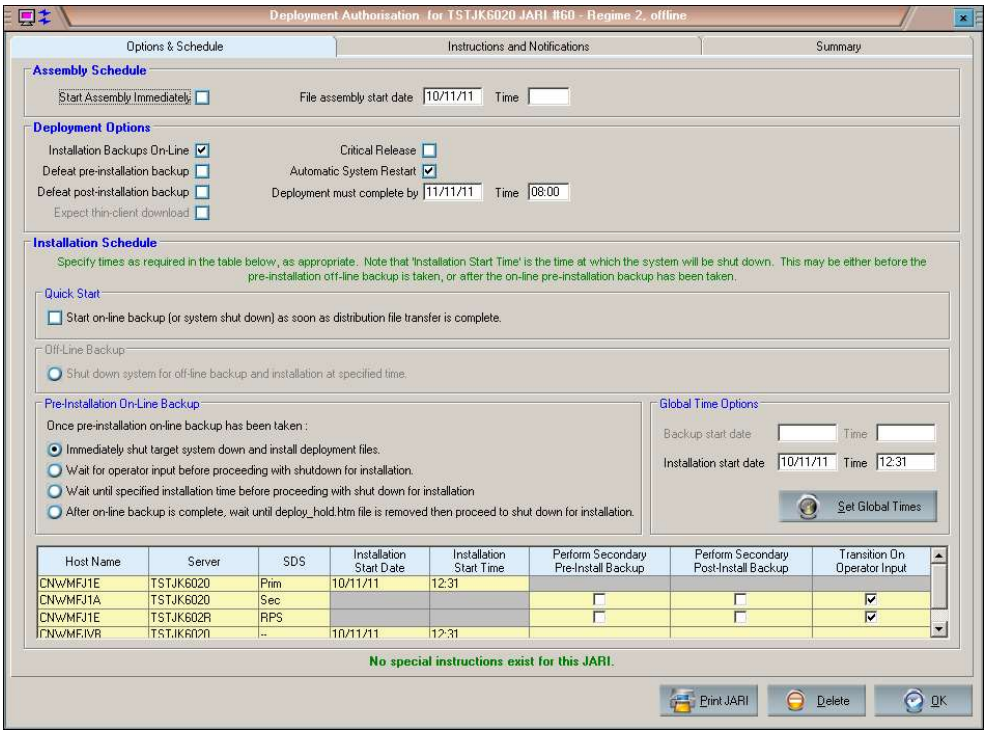
- JARI Update form – single JARI approval
- JARI Multi-Approval – for JADECare 24 only – allows approval of multiple JARIs by Environment Type.

JARI Update Form

The following form is used to:

- Authorize the deployment
- Provide scheduling information (assembly, backup, installation, transition).
- Provide any special instructions (if any)

Note that if deployment defaults have been defined for the Master Environment, these defaults will be used to populate this form.



Control descriptions:

Assembly Schedule	The option exists of scheduling the assembly for a specified date and time OR of starting the assembly as soon as this JARI authorization is completed
Deployment Options	
Installation Backups On-Line	This is on by default. Allows the authorizer to specify online or offline backups. If offline, then the system is taken down before the

	backup and proceeds directly into the installation.
Defeat pre-installation backup	This avoids a pre-deploy backup being taken. However, if the installation aborts then manual restore procedures will be required.
Defeat post-installation backup	This avoids a post-deploy backup being taken.
Critical Release	Documentation only. Used by help desk staff (if managed) to identify if call out/paging required should deployment have problems.
Automatic System Restart	Default is true. When true, the system is restarted after deployment. If false, system is left down after deployment. Note that if backups requested to be on-line (the default) and post-deploy backup not defeated, the backup database files will be taken from the pre-deploy backup and the remaining files from the host environment (including all journals required).
Deployment must complete by	Documentation only. Used by help desk staff.
Installation Schedule	
Quick Start – start on-line backup (or system shutdown) as soon as distribution file transfer is complete	Allows the pre-deploy backup (if not defeated) to start immediately after the distribution package has finished being sent to the target host environment
Off-line Backup – Shut down system for off-line backup and installation at specified time	Applicable only when backups are offline. Changes the deployment schedule table to specify installation time which means system shut-down time.
Pre-Installation On-line Backup	
- Immediately shut target system down and install deployment files	Once online backup has completed, immediately start system shutdown and proceed with installation
- Wait for operator input before proceeding with shutdown for installation	Once online backup has completed, suspend waiting for help desk prompt before starting offline shutdown.

- Wait until specified start time before proceeding with shut down for installation	This option allows for entry of scheduling information as to when the backup is to start and when shutdown (and installation) is to start.

The scheduling table displays one row for each host environment to be deployed to. This table allows unique scheduling requirements to be defined for each host environment being deployed to.

Bear in mind that date and time specified are based on that of the server that JSM is running on.

Global Time Options

This set of controls is only available when there is more than one database base environment (with a role of Primary or Non-SDS) being deployed to.

In that case, the date and time for backup and/or installation time for all those databases can be set to the same value and then individual entries can be further adjusted, if required, by modifying the appropriate date/time entry in the table.

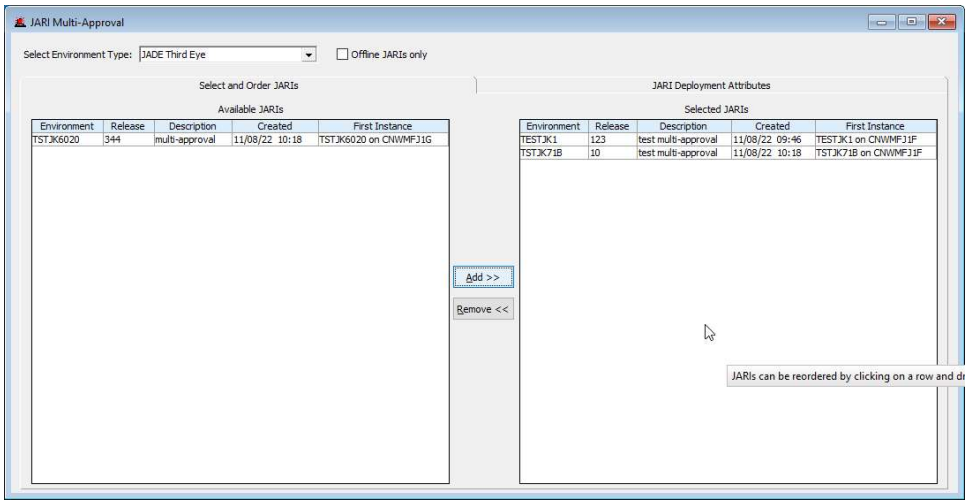
JARI Multi-Approval

The use of this form is restricted to JADECare 24 only.

The following form is used to:

- Select JARIs by environment type and whether online or offline deployment
- Be able to order the environments into priority sequence.
- Provide scheduling information (assembly, backup, installation, transition).

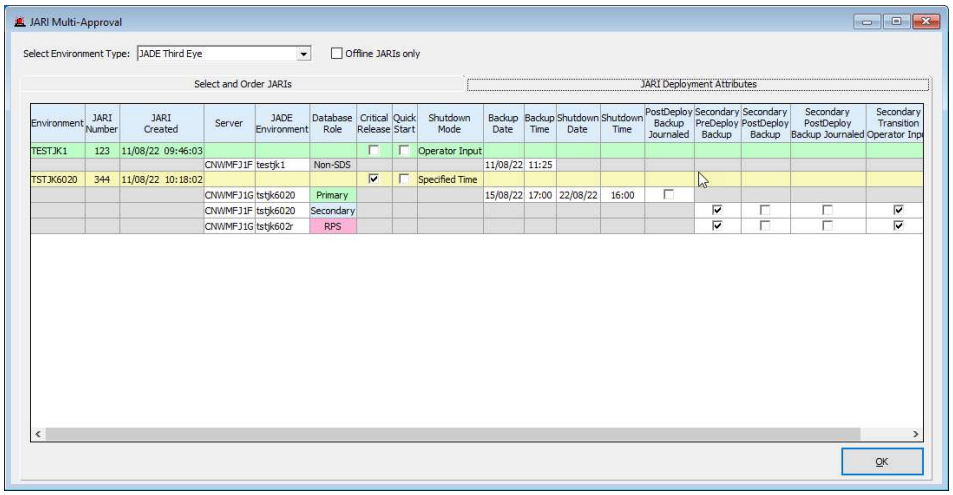
Note that if deployment defaults have been defined for the Master Environment, these defaults will be used to populate the relevant items.



Select Environment Type	Select one of the available environment types. This will cause the Available table to be populated with JARIs meeting the criteria.

Offline JARIs Only	When checked (the default) this will restrict JARI selection to those that are offline deployments only. When not checked the selection is restricted to online deployments only
Available JARIs	JARIs can be selected by selecting multiple rows and using the Add button. JARIs can also be add to the selected table double clicking on an entry in this table
Selected JARIs	JARIs can be removed from the section by selecting multiple rows and clicking the Remove button. JARIs can also be removed from the selction by double-clicking on an entry in this table. The JARIs can also be re-ordered in this table by selecting a row and dragging it into the desired position.

The selection and re-ordering process can be repeated as often as required. When user clicks on the **JARI Deployment Attributes** sheet this will populate / update the table shown in the order of the JARIs displayed in the **Selected JARIs** table.



Each JARI is displayed with some options provided and below each are the relevant details of the JADE Environments associated with the Master Environment.

Where an environment has deployment defaults enabled, this is shown by the JARI background colour being a pale yellow and the options provided are derived from the environment options.

Note that when the OK button is clicked, all of the JARIs present have the scheduling details validated and an error for any one will cause an error to be raised and none of the JARIs will proceed to the authorisation stage.

When all validation is correct, then the first JARI is scheduled to start assembly in 2 minutes time and each subsequent JARI will have assembly scheduled at

intervals defined via the **System Global Options** value **JARI Multi-Approval Staging Interval.**.

JARI Validation Checks

Whena JARI is built the JADE and CardSchema version of the environment is retained with the JARI. This allows the JARI to check at authorization that the JADE environments have not changed (ie as the result of a restore) between the time of a JARI build and its subsequent authorization.

Using JSM's ini file settings [JSMDeployment] JariEnhancedVersionCheck=true allows this basic check to be extended. Refer to details of this ini file setting earlier.

Monitoring Deployments

There are two distinct stages that a deploy goes through

- Assembly
- Installation

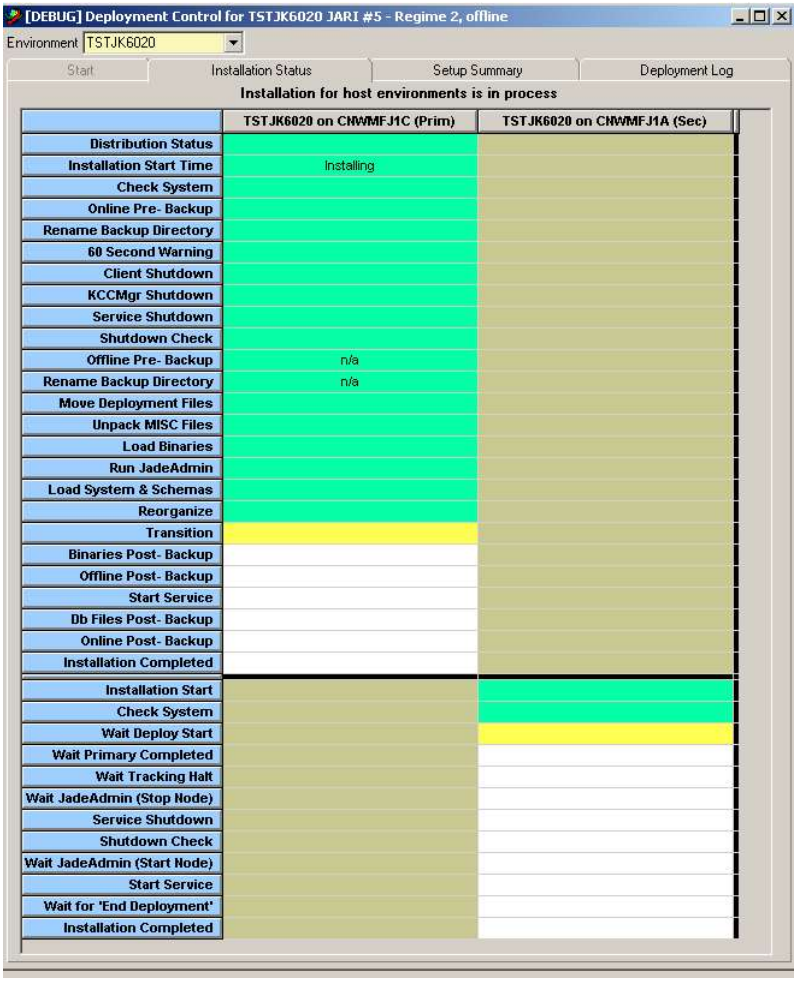
The person authorising the JARI (see JARI Wizard above) can schedule both of these stages. Central Systems can override the schedule start times and force one or both of these stages to start immediately.

Both of these stages can be monitored and actioned via the Deployment Control form (see Installation later in this section).

Assembly

This is the process of taking each of the different sets of files, creating versioning files, bundling these up and shipping them to the appropriate staging area on the host concerned via JSA transfer process ready for installation to begin.

Installation



An installation may be one of the following two types:

- Online Deployment
- Offline Deployment

The example above shows an Offline Deployment for a Master Environment that has an SDS Primary host environment with one SDS Secondary associated with it.

Each deployment type has a number of distinct states which may be monitored by the Deployment Control (see picture above in this section). Some of the states are optional, like the execution of scripts only occurs if the scripts are present, the JADE upgrade only occurs if a JADE version upgrade, JADE patch consolidation or JADE hotfixes are included in the deployment package.

A description of the colours is provided via a menu option **Display Legend**, see menu items below.

Pre-Deploy Backup

If an installation is aborted after the pre-deploy backup was successfully taken, then when the installation process is restarted and the pre-deploy backup is still considered current, then the pre-deploy backup will NOT be redone.

For the pre-deploy backup to be considered current, this means that the last files had to be copied within the last 30 hours.

Post-Deploy Backup

The post-deploy backup can be online or offline and invokes the database backup methodology used for standard and adhoc backups.

Menu Items

A selection of menu items is provided when you click on a column within the assembly or installation table. These options are enabled /disabled dependent on the current deployment status of each host environment and whether or not they are part of an SDS environment.

Show Bubblehelp

This item may be checked or unchecked and the value is retained for each user across sessions. If checked, then each time the mouse rolls over the sheet or table control the relevant bubble help will be displayed.

Display Legend

This pops up a dialog which explains the various colours used in the Deployment Control form.

Start Installation

Allows the user to start the installation immediately on the selected server host regardless of any scheduled time.

Change Installation Times

This allows the various times for a deployment on a specific host environment to be modified after the files have been distributed for that host environment.

Distribute Deployment

As most deployments automatically initiate the deployment files once the assembly phase is completed, this option will only be available if the distribution failed for some reason.

Examine Status

There are various stages through a deployment where the deployment may halt and require operator intervention. Some of these may have been specified halts as part of the JARI authorisation and SDS options, others may be issues arising as the deploy progresses. This option allows the operator to interrogate the reason for the halt.

NOTE: Any cell displaying this status may also be double-clicked on and the status will also be shown.

Continue Installation

Causes a suspended installation to continue.

Pre-Deploy Backup Progress

If a pre-deploy backup is in progress (online or offline), selecting this menu option will cause a backup progress dialog to be displayed which provides detailed overview of the backup's progress.

Double-clicking on such a cell while the backup is active will provide the same display.

Post-Deploy Backup Progress

If a post-deploy backup is in progress (online or offline), selecting this menu option will cause a backup progress dialog to be displayed which provides detailed overview of the backup's progress.

Double-clicking on such a cell while the backup is active will provide the same display.

Note: the binaries post-backup and db files post-backup progress may **not** be viewed as this is a hybrid backup scenario for which backup progress is not reported to JSM.

Restore Progress

If a restore is in progress, selecting this menu option will cause a restore progress dialog to be displayed which provides a detailed overview of the restore's progress.

Double-clicking on such a cell while the restore is active will provide the same display.

Abort Installation

This causes the installation on this specific host environment to abort and to restore to the starting point. The behaviour for environments that are SDS secondaries is quite different to that of SDS Primary and non-SDS environments

Kill Deployment

This causes **ALL** host environments for this master environment to be aborted and restored and then leaves the JARI in a 'Requires Rescheduling' state.

Note that this option is no longer available when any SDS secondary environments are party to the deploy and when any one SDS Primary has achieved transition. The recovery scenario in such a case is totally manual and is up to the operations team as it requires knowledge of the SDS environment(s) concerned and the state it is currently in.

Mark JSA Deploy Complete

Applies to Jade System Agent deployments only.

This causes the deployment to a particular JSA to be marked as completed. Typically, this would be used to enable manual completion of a deployment where one or more of the JSAs is offline (was unable to be deployed to).

When this option is selected, the operator is responsible for ensuring that the physical deployment to the JSA is complete and that version files are consistent with those held in JSM at the end of deployment to all JSA environments.

Deployment States

The following states may occur more than once in the case of a nested JARI, each of the state descriptions will be qualified by the JARI type (parent or child) and the JARI release number.

- Unpack MISC files
- Script 'live_preload'
- JADE Upgrade
- CardSchema Upgrade
- Load Binaries
- Script 'live_preload2'

- Load System and Schemas
- Load System and Schemas
- Read-Only Schema
- Script 'live_postload'

Offline Deployment :

An offline deployment is one in which all changes are applied to the database and any reorganisations are done while the system is offline. The states that can apply to an offline deployment are as follows:

Online Deployment :

An online deployment is one in which all the changes are applied while the database is online, with the database being versioned. This allows usage of the database using the same logic and object properties as at the start of the deployment. After the transition stage (in which all the changes are implemented), the system must be taken down for a very brief period in order for these changes to be applied at the application level. This type of deployment aims to keep the system downtime to an absolute minimum.

Check System

Identify the current status of the target host environment and execute pre-deployment checks.

Online Pre-Backup

If the system is running and the **Installation Backups On-Line** option was checked in the JARI authorisation/update form, then the installation process will execute the JSM Managed Database Backup. The backup directory used is the **Pre-Installation Backup Directory** specified for the Host Environment, the backup options are online and verify.

Before the backup actually starts, if the file **live_prebackup.cmd** is resident under the **server\c_misc\cardinal** directory, it will be executed.

While this phase is showing as active (yellow), double clicking on this cell will cause the Backup Progress form to be displayed, this giving a tabular and graphical display of the backup states.

If this phase did not execute, then it will display **n/a** (not applicable) even though the status will show as green.

For details on the JSM Managed Database Backup refer to the document Database Backup – Phase 1 Disk to Disk version 2.7.100.

NOTE: The Backup Schedule is updated with installation initiated backups.

Rename Backup Directory

The renaming of the interim backup directory to its final name only occurs once a successful backup is obtained. This state may require a retry if the rename does not succeed (eg if a file is in use).

60 Second Warning

If the file **live_pre_systemdown.cmd** is resident under the **server\c_misc\cardinal** directory, it will be executed.

All clients of the target host environment are sent a warning message that the host environment will shut down in 60 seconds

Client Shutdown

Waiting for 60 secs and send shutdown messages to all clients requesting terminate

KCCMgr Shutdown

send KCCMgr of host environments a message to forcibly terminate any remaining clients and to terminate itself

Service Shutdown

Send commander request to JAWS for service to be shut down

Service Shutdown

Check that database is offline.

Offline Pre Backup

If the **Pre-Installation Backup On-Line** option was not specified or the system was already down when the installation started, the installation will execute an offline backup.

Before the backup actually starts, if the file **live_prebackup.cmd** is resident under the **server\c_misc\cardinal** directory, it will be executed.

The database and all other environment files are backed up into the **Pre-Installation Backup Directory** defined for this host environment. The database backup is done using the JADE utility **jdbutilb**.

While this phase is showing as active (yellow), double clicking on this cell will cause the Backup Progress form to be displayed, this giving a tabular and graphical display of the backup states.

Move Deployment Files

Moves the version control files and deployment files from next directories into current directories

Unpack MISC files

Unpack miscellaneous files.

Script 'live_preload'

Execute script **live_preload.bat** present

JADE Upgrade

starting JADE upgrade (version, patch consolidation and/or hot fix(s)). Uses utility **jadloadb** to load any RootSchema or JadeSchema patches and moves JADE binaries into the binary directory.

CardSchema Upgrade

starting CardSchema upgrade using utility **jadloadb**.

Load Binaries

starting load of binaries

Run JadeAdmin

On a primary, this process creates an updated manifest file which is to be used by fat-clients (app servers, JADE.exe) and by SDS secondary environment to identify changes to the binaries and other required files.

Script 'live_preload2'

Execute script **live_preload2.bat** if present

Load System and Schemas

If 'Certify Database Before Load' option selected in the JARI Wizard, then run utility jdbutilb with command **certify** before schema changes are applied.

Load schema changes provided by user using utility **jadloadb**.

Reorganize

Forces the various map files to be modified if a reorganization was required and causes the latest method changes to become current methods.

Script 'user_pretransition.bat'

If online deployment, execute script **user_pretransition.bat** if present

Transition

Commits all the changes made and is effectively the point at which they become effective.

Script 'live_app_upgrade'

run script **live_app_upgrade** if present.

Read-Only Schema

If the option 'Uses Read-Only Schema' is set for the Master Environment, then extracts and update files required to implement this feature.

Database Certify

If the option 'Certify Database After Load' was selected in the Jari Wizard, then before the database is restarted and after all changes have been implemented, the database is certified by running utility **jdbutilb** with command **certify**.

Script 'live_postload'

Execute the script **live_postload** if it is present.

NonDb Post- Backup

This occurs if:

- Post-install backup has not been defeated
- A reorg did not occur
- The pre-install backup was not defeated

This state causes all the non-database files, included directories and those not specifically excluded to be copied to the **Standard Backup Directory**.

Double clicking on this cell will NOT cause the Backup Progress form to be displayed as this type of backup is not a typical JSM managed backup.

If this phase is not actually executed then the text **n/a** is displayed in this cell.

Offline Backup

This state is invoked if:

- Post-installation backup was not defeated
- Pre-install backup was defeated
- Offline backup was requested
- Reorg occurred and ini file option **UseReplayableReorg** was not found or not set to true.

The backup is of all files (including the database), included directories and excludes those defined in the exclusion list, these being directed to the **Standard Backup Directory** defined for this Host Environment.

While this phase is showing as active (yellow), double clicking on this cell will cause the Backup Progress form to be displayed, this giving a tabular and graphical display of the backup states.

If this phase is not actually executed then the text **n/a** is displayed in this cell.

Start Service

This is invoked when the **Automatic System Restart** is selected in the JARI Wizard or selected and overridden by the JARI Authorization form.

DB Files Post-Backup

This state is invoked if the **NonDB Post-Backup** state was invoked. It copies the database files from the pre-installation backup directory into the **Standard Backup Directory** defined for this Host Environment.

Double clicking on this cell will NOT cause the Backup Progress form to be displayed as this type of backup is not a typical JSM managed backup.

If this phase is not actually executed then the text **n/a** is displayed in this cell.

Online Post Backup

Invoked if:

- Post-installation backup was not defeated
- Database backup was not specifically requested to be offline
- Reorg was detected and the ini file option **UseReplayableReorg** was set.

Starts an online backup of all files into the **Standard Backup Directory** defined for this Host Environment.

While this phase is showing as active (yellow), double clicking on this cell will cause the Backup Progress form to be displayed, this giving a tabular and graphical display of the backup states.

If this phase is not executed, then the text **n/a** is displayed in this cell.

Script 'live_postbackup'

If the script **live_postbackup.**cmd is located in **server\c_misc**cardinal directory then it will be executed.

Required Files for Scripts

The following deployment scripts will often require specific files to accompany them as part of an online deployment:

- live_preextract.bat
- live_preload_online.bat
- user_pretransition.bat

As these required files will not have be available at the time these scripts are to execute (as they have not been unpacked) the following methodology has been provided to be able to specify these files as part of the script concerned.

If one or more files are required for either of the above scripts then:

- They must be bounded by literals:

- <BeginRequiredFiles>
- <EndRequiredFiles>

- they are to be named relative to the directory they reside in
- they may be exist on one or more sub-directories
- they may or may not be preceded by the 'rem ' comment statement

The required files are defined within the script, then each file named must be present in the same directory as their respective scripts

Example:

If the following script is called 'live_preload_online.bat' and the script is resident under directory d:\deployments\MyExampleEnv\database\miscFiles

```
echo on
set var1=testing
<BeginRequiredFiles>
testfile1.txt
subdir1\testfile2.txt
subdir1\subdir2\testfile3.txt
<EndRequiredFiles>
#runjade schema=TestSchema app=TestApp server=singleUser
```

then the following files must be located as follows:

d:\deployments\MyExampleEnv\database\miscFiles\testfile1.txt
d:\deployments\MyExampleEnv\database\miscFiles\subdir1\testfile2.txt
d:\deployments\MyExampleEnv\database\miscFiles\subdir1\subdir2\testfile3.txt

These files will also be removed before JadeAdmin is executed so they are not propagated to SDS Secondaries or remote fat clients.

User Deployment Hooks

During the various phases of a deploy, there will be times where a batch script needs to be executed. The online backup process has its own 'user hooks', for details (refer to the documentation on 'JSM Managed Backup').

The scripts to be provided will have pre-defined variables and error checking wrapped around them, see 'Items Provided' below

The following deploy specific batch scripts may be released as part of a deploy and will be executed if found to be resident and then renamed unless otherwise stated or the keyword 'dontrenameafterrun' is not found anywhere in the script:

If one or more of the following scripts are to be executed, they may be provided as part of the deploy under the <database distribution> | <characterset> | Server | Misc Files selection in the Jari Wizard or in the directory specified in the <database distribution>ServerMisc item in the Batch JARI file.

'live_preextract.bat'

This script is extracted and executed (if deployed) in both online and offline deployment variants. It is executed when the JADE Environment is offline and immediately before the deployment package extracts all the binary and other associated files.

Its primary purpose is to allow developers to delete unwanted files and /or directories before the deployment package content is extracted.

‘live_preload_online.bat’

This script is executed at the beginning of an on-line deployment. Note that any files that are required by the script must be defined within the script such that these can be extracted and placed in the required directory prior to starting the script, i.e. the files cannot be simply included in the 'misc' distribution, refer to section above ‘Required Files For Scripts’

Also note that JADE command lines within this script that open the database need to specify 'server=multiuser', because the database will still be running at the time that the script is run.

‘live_preload.bat’

For **off-line** deployments, this is executed after the pre-deploy backup has been done, the system is confirmed as being down and the ‘Misc files’ have been deployed but before any of the other upgrade steps have occurred.

For on-line deployments, the live_preload.bat script is executed after schema file load/reorg/shutdown and transition. For on-line deployments, if you need to run a script before **anything** is deployed to the database, you should use the **live_preload_online** script.

‘live_preload2.bat’

For off-line deployments only. Executed after the pre-deploy backup has been done, once the system is confirmed as being down, the ‘Misc files’ have been deployed and after any Jade / CardSchema upgrade and new binaries have been upgraded but before application of submitted schema file changes.

'user_pretransition.bat '

For on-line deployments only. Executed after any schema files have been loaded and the system has shut down, immediately before schema transition is initiated.

Note that any files that are required by the script must be defined within the script such that these can be extracted and placed in the required directory prior to starting the script, i.e. the files cannot be simply included in the 'misc' distribution, refer to section above ‘Required Files For Scripts’

‘live_jade_upgrade.bat’

Executed as part of a Jade upgrade after the binaries and database files have been loaded but before any Jade provided schema files have been loaded.

This file is renamed ‘live_jade_upgrade.ba~’ after execution regardless of the keyword ‘dontrenameafterrun’ being specified within the script.

‘live_app_upgrade.bat’

Executed after binaries and schema loads have completed and before post-installation backup is started.

This file is renamed ‘live_app_upgrade.ba~’ after execution regardless of the keyword ‘dontrenameafterrun’ being specified within the script.

NOTE:

At the end of executing the live_app_upgrade script, the deploy process expects to find a log file titled ‘app_upgrade.log’ under the environments log directory (the log directory may be obtained from initialisation file setting LogRootDirectory in section PersistentDb).

This log file must contain at least one line with the word '**completedok'** starting from the first character. The keyword check is NOT case-sensitive.

Failure to find this log file or the keyword will cause the deploy to go to hold state allowing operations staff to either Ignore (and continue) or to Cancel (which aborts the deployment).

'live_postload.bat'

Executed after the 'live_app_upgrade.bat' is executed, if found, after binaries and schema loads have completed and before post-installation backup is started.

Operational Deployment Hooks

The following two scripts may not be provided as part of the deploy process and are provided for operational use only.

'live_prebackup.cmd'

Script will be executed if located under directory **<env>\server\c_misc\cardinal**.

This script is executed before a pre-deploy backup (online or offline) is to be done. **Note** that if the pre-deploy is considered **'current'** (see below Installation Backups) and the pre-deploy backup is skipped, this also means that this script will NOT be executed as it will have been executed at the time the previous pre-deploy backup was taken.

An offline pre-deploy backup will only be done when:

- the deploy was started with offline backup specified
- the system is detected as not running when the deploy starts

'live_pre_systemdown.cmd'

Script will be executed if located under directory **<env>\server\c_misc\cardinal**.

The script is executed when the shutdown time has passed and before the deployment issues a 60-second warning of shutdown to registered application clients.

'live_postbackup.cmd'

Script will be executed if located under directory **<env>/server/c_misc\cardinal**.

The script is executed after the system service has been restored and after the post-installation backup has been taken.

'start_of_deploy.cmd'

Script will be executed if located under directory **<env>/server/c_misc\cardinal**.

The script is executed after the pre-deploy backup phase has completed and immediately before the deployment archive files are extracted.

Deployment Scripts

Command Line Name

Where the host environment database node was started with a 'Name=<command line name>' option so as to use ini file named sections, this same name is inserted into all command line strings where an executable is executed. The insertion point is deemed to be after the first space located after the word 'ini'.

Items provided

NOTE: The variables listed below are also provided in the BATCHRUN command facility provided by **Commander** however, the keywords below and the ERRORLEVEL 1 checks inserted into the script are not provided nor will they be supported by **Commander.**

Several variables and a keyword are provided by JSM in order to simplify the creation of the above scripts.

Keyword 'no_check'

Any executable will normally be immediately followed by an errorlevel 1 test which will cause the script to be deemed to have failed. However, this test may be overridden by including the 'no_check' as the last entry on the line to be executed.

Eg:

```
%env_binpath%\jadloadb ini=%env_binpath%\%env%.ini
path=%env_systempath% schemaFile=%env_miscpath%\schema1.scm
ddbFile=%env_miscpath%\schema1.ddb no_check

%env_binpath%\jadloadb ini=%env_binpath%\%env%.ini
path=%env_systempath% schemaFile=%env_miscpath%\schema2.scm
ddbFile=%env_miscpath%\schema2.ddb
```

will generate a script as follows:

```
%env_binpath%\jadloadb ini=%env_binpath%\%env%.ini
path=%env_systempath% schemaFile=%env_miscpath%\schema1.scm
ddbFile=%env_miscpath%\schema1.ddb

%env_binpath%\jadloadb ini=%env_binpath%\%env%.ini
path=%env_systempath% schemaFile=%env_miscpath%\schema2.scm
ddbFile=%env_miscpath%\schema2.ddb
if ERRORLEVEL 1 then goto _NOTOK
```

The first jadloadb loads schema and ddb title "schema1" and no error check is done, whereas the second jadloadb loads schema and ddb "schema2" and will be checked for any errors.

Keyword '#runjade'

In order to run an application within the target system, the keyword '#runjade' (not case-sensitive) will insert the qualified jade.exe including the path and initialisation file references, add the schema and app specifications from the script provided and append the appropriately derived 'server=SingleUser|MultiUser' and any 'endJade' parameters.

Note: Any other details preceding the 'endJade' keyword (such as ini file parameters) are ignored.

Example:

Script statement for environment TESTJK25 executing from C drive

```
#runjade schema=TestJK app=ConvertForUpgrade endJade <parameters>
```

will be replaced with:

c:\testjk25\server\c_bin\jade.exe path=c:\testjk25\server\c_system ini=c:\testjk25\server\c_bin\testjk25.ini schema=<provided by script> app=<provided by script> server=%_server_value% name=<commandline name> endJade <parameters>

Keyword '#runjadclient'

In order to run an application within the target system, the keyword '#runjadclient' (not case-sensitive) will insert the qualified jadclient.exe including the path and initialisation file references, add the schema and app specifications

from the script provided and append 'server=SingleUser|MultiUser' and any 'endJade' parameters.

Example:

Script statement for environment TESTJK25 executing from C drive

```
#runjadclient schema=TestJK app=ConvertForUpgrade endJade <parameters>
```

will be replaced with:

c:\testjk25\server\c_bin\jadclient.exe path=c:\testjk25\server\c_system ini=c:\testjk25\server\c_bin\testjk25.ini schema=<provided by script> app=<provided by script> server=%_server_value% name=<commandline name> endJade <parameters>

Keyword 'dontrenameafterrun'

any script executed will automatically be renamed by appending the date and time (format YYMMDDHHMMSS) to the file name unless the above keyword is found anywhere in the text.

Variables provided

env	the environment name
env_drive	the drive letter for this environment
env_dir	the base directory for this environment
env_systempath	the path for this environments system files
env_binpath	the path for this environments binaries
env_binpath32	the path for this environments 32-bit binaries. If 32-bit binary path doesn't exist, then the value will be the same as **env_binpath**
env_miscpath	the path for this environments miscellaneous
env_logpath	the path for this environments log files
env_altospath_client	the path for this environment's altos client files
env_altospath_server	the path for this environment's altos server files
env_altospath_thinclient	the path for this environment's altos thinclient files
env_inifile	the path and name of the initialization file
env_temp	the path for this environments temporary release directory
env_characterset	the character set of the environment: ansi or unicode
env_devdb	identifies if a development or deployment only environment 0 – deployment only 1 = development
env_reportwriter	identifies if ReportWriterSchema is part of this environment 0 – report write not available 1 – report writer is available
env_dbActive	identifies if JSA saw environment as active (database running with KCCMgr) 0 – database is not active 1 – database is active

env_deployregime	deployment regime for master environment (retained for compatibility and future use)
env_backuptype	identifies if executed from backup process and what kind unknown – unable to evaluate notapplicable – script not started by backup process scheduled – scheduled backup adhoc – adhoc backup, immediate or scheduled predeploy – pre-deployment backup postdeploy – post-deployment backup
env_logdays	the number of days to retain logs, aka 'Database Log Days'
env_backupmode	identifies the backup mode , values can be offline, online journaled.
env_databaserole	the current database role for the host environment
env_jadeversion	last registered JADE version for host environment eg 6.3 – JADE 6.3, 7.0 – JADE 7.0
env_workersoffline	number of offline backup workers defined
env_backuppath	the directory to which the backup will be directed. This is only relevant for scripts associated with the backup process.
env_distribution_count	Installation scripts only. Gives the number of distributions in this deployment.
env_distribution<n>	Installation scripts only. Defines one of the deployment.distributions
env_emailaddress	Developer Email Address as defined for the JADE Environment in JSM.
env_fmsassignee	FMS Assignee as defined for the JADE Environment in JSM.

Example of script

The examples below show what may be supplied by a developer and what will actually executed by the deployment process:

Developer supplied:

```
rem % Batch file for execution of application during deploy  %

% env_binpath %\jade.exe path=%env_systempath% ini=%env_inifile%
schema=TestJK app=DeployScriptOne

#RunJade schema=TestJK app=DeployScriptTwo

if %env_reportwriter% NEQ 1 goto :next

#RunJade schema=ReportWriterSchema app=FictitiousApp

:next
```

Deploy actually executes:

```
echo on
set env=testjk25
set env_drive=C
```

```
set env_dir=C:\testjk25
set env_systempath=C:\testjk25\server\c_system
set env_binpath=C:\testjk25\server\c_bin
set env_binpath32=C:\testjk25\server\c_bin32
set env_logpath=C:\testjk25\logs
set env_inifile=C:\testjk25\server\c_bin\
set env_temp=C:\testjk25\server\c_system\release_temp
set env_characterset=ansi
set env_databaserole=1
set env_deployregime=3
set env_backuptype=scheduled
set env_devdb=1
set env_reportwriter=0
set env_dbActive=0
set env_convertedControlFile=0
set env_jadeversion=6.3
set env_workersoffline=2
set env_installmode=online
set env_backuppath=f:\cnwbackup_temp\testjk25
set env_distribution_count=2
set env_distribution1=i686-msoft-win32
set env_distribution2=x64-msoft-win64
set env_emailaddress=testemail@somedomain.co.nz
set env_fmsassignee=some_csr
set env_lastVariable=999

rem % Batch file for execution of application during deploy  %

C:\testjk25\server\c_bin\jade.exe path= C:\testjk25\server\c_system
ini= C:\testjk25\server\c_bin\testjk25.ini schema=Testjk
app=DeployScriptOne

if ERRORLEVEL 1 goto :_NOTOK

if %env_dbActive% NEQ 0 goto :_ACTIVE1

c:\testjk25\server\c_bin\jade.exe schema=TestJK app=DeployScriptTwo
path=c:\testjk25\server\c_system
ini=c:\testjk25\server\c_bin\testjk25.ini server=singleuser
if ERRORLEVEL 1 goto :_NOTOK

goto :_CONTINUE1

:_ACTIVE1
c:\testjk25\server\c_bin\jade.exe schema=TestJK app=DeployScriptTwo
path=c:\testjk25\server\c_system
ini=c:\testjk25\server\c_bin\testjk25.ini server=singleuser
if ERRORLEVEL 1 goto :_NOTOK

:_CONTINUE1

if %ENV_REPORTWRITER% NEQ 1 goto :next

if %env_dbActive% NEQ 0 then goto :_ACTIVE2
c:\testjk25\server\c_bin\jade.exe schema=ReportWriterSchema
app=FictitiousApp path=c:\testjk25\server\c_system
ini=c:\testjk25\server\c_bin\testjk25.ini server=singleuser
if ERRORLEVEL 1 goto :_NOTOK

goto :_CONTINUE2

:_ACTIVE2
c:\testjk25\server\c_bin\jade.exe schema=ReportWriterSchema
app=FictitiousApp path=c:\testjk25\server\c_system
ini=c:\testjk25\server\c_bin\testjk25.ini server=multiuser

if ERRORLEVEL 1 goto :_NOTOK

:_CONTINUE2
```

```
:next

rename c:\karma271\logs\i_testjk25_script.run i_testjk25_script.end
goto :_END
:_NOTOK
rename c:\karma271\logs\i_testjk25_script.run i_testjk25_script.err
:_END
```

If it is required to return a non-zero exit code to the script (e.g. when using the #RunJade command) then your JADE application should assign the desired exit code value to **Node::userExitCode** before terminating. When the node terminates, this exit code will be passed to the script's ERRORLEVEL value.

User Schema Load Scripts

User schema load scripts are applicable to OFFLINE deployment only. When scheduling a deployment a typical scenario may deploy:

```
Test1.scm
Test1.ddb
Test1.jcf
```

Once this is deployed the schema load order would load the .jcf file first followed by the .scm and .ddb file. A user can provide an additional script to be executed after the schema loads have been completed successfully. In this example:

```
Test1.scm
Test1.ddb
Test1.jcf
Test1.bat
```

Test1.bat will be executed after the .scm and .ddb have been successfully loaded. The naming convention for the .bat files needs to be the same as the load file being submitted. For instance, submitting:

```
Test1.scm
Test1.ddb
Test1.jcf
Test2.bat
```

will cause an error as Test2.bat has no schema load file counterpart. In addition, no duplication of filenames is permitted, for example:

```
Test1.scm
Test1.ddb
Test1.jcf
Test1.mth
Test1.bat
```

will cause an error as the user provided script name is the unique identifier to bind it to the target schema load file. In the case above this matches both the .scm + .ddb file but also matches the .mth file.

The exception to this rule is the combination of .jcf, .scm + .ddb files. If a .jcf file is being provided the contents of the .jcf file are examined to make sure it is bound to the provided .scm file. If it isn't, an error will be raised as this would be considered a duplicate schema load filename.

A .jcf file, or for that matter any recognised load file, can be submitted on it's own with its own load script. Multiple files with multiple scripts can also be submitted providing the naming standard is adhered to.

Script Submission

Scripts should be submitted for deployment from the same location as the schema load files.

Script Runtime

Once the script has been successfully executed the script will be removed from disk, this is to prevent accidental re-running of a script.

Given only the person submitting the script has understanding of the scripts task, this is the safest option. If the file cannot be removed from disk it will result in a deployment failure.

Script Execution Failures

User schema load scripts will cause a deployment failure if they fail to execute properly, in this case a restore or manual intervention will be required.

JadeAdmin, Autodist, cnwplugin Automatic Updates

As from the 4.1.01 release, each deployment will now check that the following deployed code files have the same version as those registered for the base release defined for that environment:

- jadeadmin.exe
- autodist.dll
- cnwplugin.dll

If the versions differ, then the above code files will be unconditionally deployed (including corresponding pdb files) for the distributions applicable.

Environment Shares

During a deployment, once the system is taken offline, any cluster file shares defined for the environment (and its sub-directories) will be taken offline and any open Window file shares will be closed. This also applies to any inclusion backup directories declared for the environment regardless of whether a pre-deploy backup was requested or not.

When the offline backup is completed, any cluster shares taken offline will be brought back online.

NOTE: The action of taking cluster file shares offline and online may generate SNMP traps which will shows as alerts within JSM.

Installation Backups

WARNING:

If a pre-deployment backup was taken and completed within the last 30 hours, then the install may be restarted multiple times and the pre-deploy backup will NOT be redone as the earlier one will suffice. The required transaction journals for recovery are copied into this backup at the time the environment is stopped.

The backups taken during the deployment process use the same mechanism as the JSM Managed Backup (the only exception being the post-deploy backup where the backup is split into two distinct phases, see below). The deployment process always ensures that two backups are taken during the installation phase. These are known as:

- Pre-Installation Backup
- Post-Installation Backup

The pre-install backup represents the system before any software or logic changes are applied.

The post-install backup represents the system after all changes have been implemented.

Pre_Install Backup

The pre-install backup will be online if all the following are true:

- The JARI being authorised specifies online (the default)
- The target system at install time is active
- The target system has a CardSchema version that caters for online backup (2.7.01 or higher)

Post-Install Backup

The post-install backup will be a full offline backup if any of the following are true:

- The JARI specified the backup to be offline
- The changes to the system caused a reorganization
- One or more .dat files were added to the system path (Jade version upgrade)
- The host environment is not set up to use replayable reorgs. Note : ini file setting UseReplayableReorg should always be true for managed systems using JADE 6 and beyond.

If the post-install backup is not offline, it is split into two parts. The first part ensures that all the non-database files (which may have changed as a result of the deploy) and latest database transaction logs are copied. Then system is then restarted (unless specifically advised not to in the Jari wizard). The database files from the pre-install backup are copied into the post-install backup directory.

Abort Installation

Refer also to '**Kill Deployment**'.

The 'Abort Installation' menu item causes the install phase of the deploy to be aborted, leaving the host environment in a state where the install can be restarted (the assembly and distribution phase do not need to be repeated).

Once the installation phase has moved into or past the Upgrade phase, then an abort request will cause the host environment to be restored from the pre-deployment backup.

NOTE: If a pre-deployment backup was taken, then as long as the host environment is NOT started, restarting the install means that the pre-deploy backup will NOT be redone as the earlier one will suffice.

Once the host environment has been taken offline, if an abort is executed before the environment is restarted then the abort action will NOT restart the system, this is to prevent environmental flags in JSA from being reset, which would invalidate the pre-deployment backup.

If a pre-deployment backup was not done (defeated), then if the deployment had started any physical changes (the Upgrade and Install phases), then no restore is possible. At this juncture, a 'Kill Deployment' with NO option should be executed as only manual intervention will restore this system.

Kill Deployment

Refer also to '**Abort Installation**'.

The Kill deployment option comes with two variants:

1. Abort and restore host environment to pre-assembly state.
2. Reset the specific host environment but do not restore it and does not affect the MASTER environment.

Option 1 effects a restore similar to that of the 'Abort Installation' where the host environment is restored (either from pre-deployment backup or via 'abortReorg' and 'unversion schemas') but goes further in that the distributed files are deleted requiring the assembly and distribution phases to be redone.

Option 2 leaves the host environment in the state at which the deployment was aborted. It normally requires manual intervention when this option is used to restore the host environment. In order to reset the environment flags within JSM and JSA in order for further deployments to work, the system should be initiated so that its registration with JSM causes the environment flags and version info to be resynchronised.

Option to Resume a Recovery after an Exception in File Copy

Database files are restored prior to non-database files.

When an exception occurs during phase two of an install (restore of non-db files), the deploy progress will go to hold state (rather than error state) and a retry/cancel dialog will be displayed.

Inspect the cause of the exception in JSA logs (for example a file was unable to be copied) and resolve as appropriate, then select the retry option in the deploy progress form. Restore will then resume at phase two (restore of non-db files).

Recovering Using Backup

If, after a successful deployment, the system needs to be recovered from the pre-installation backup directory or from an earlier backup, then any hot fixes deployed between the time of the backup and the recovery point will need to be explicitly reapplied either manually or via the JSM deployment process. These updates should be applied after the system has been restored, ideally before the system is allowed to resume normal processing.

IGNORE, ABORT, RETRY, CONTINUE states

When the install process is in this state, as long as the system is down and not running, the server host may be re-booted and other work done and as long as the same JSA (the one that has the state information for this installation) is restarted, the Install process may be restarted with the appropriate command.

NOTE: If a pre-deployment backup was taken, then as long as the host environment is NOT started, restarting the Install means that the pre-deploy backup will NOT be redone as the earlier one will suffice.

Logs and Exception Handling

TBD

Implementation of Deployment Regime 3

Defining Regime 3

Before this regime can be defined, the following criteria must have been met.

- JSA must be running JADECare Start 3.4.02 and JADECare Systems Manager 3.4.02
- All host environments for the master environment must be running
  - JADE 6.2.18 or later JADE 6.2.version
  - JADE 6.3.04 or later JADE 6.3 version
  - JADE 7
- the thinclient files already distributed need to be identified and a copy retained in the JSM staging area for each master environment.

The Environment Maintenance form has been changed and has a new sheet to display the Master Environments thinclient files. Initially this will have no data until the button provided 'Retrieve Thinclient Files' is clicked.

When the button 'Retrieve Thinclient Files' is clicked, JSA will copy back all thinclient files for the necessary distributions into local JSM directory **<drive>:\<master environment name>\thinclient\current**. The sheet titled **Thin Client Details** has 3 sub-sheets and the one titled **Current** shows the currently deployed thinclient files (and for JADE specific files, identifies any that should or should not be listed)

The thinclient files must have been retrieved within the last day before deployment regime 3 can be set. Please check the validity of the thinclient files retrieved before setting regime 3.

Deployments Using Regime 3

When creating a new JARI via the JARI Wizard, if any hot fix is applied that will cause a thinclient download, the JARI author is advised.

When a JARI is submitted, either via JARI Wizard or JARI Batch Build, the author has the opportunity to specify whether or not a thinclient download is expected. The default is the a thinclient down is NOT expected.

The option to set or reset the 'Expect thinclient download' option is also provided in the JARI Update form when the JARI is authorised and scheduled.

The contents of directory **thinclient\current** are copied as **thinclient\next** and JADE release base files, JADE hotfixes, CardSchema files and and user files

defined in the JARI are added to or overwrite existing files in the **thinclient\next** directory.

It is only during the first phase of assembly that JSM can detect whether or not a thinclient download will or will not occur. If there is any conflict between the 'Expect thinclient download' option on the JARI and what JSM detects, then the assembly aborts. If a thinclient download is detected and the option was not specified, then the files changed, added or deleted are named in the error message and deployment log.

The contents of the **thinclient\next** directory form the basis of what is deployed to the database host environment and the utility **JadeAdmin** (via **cnwplugin**) transfers these files to the appropriate **altos\client\thinclient** directory structure.

While the deployment is under way, it is possible to view the Current and Next thinclient files (once phase one of the assembly process has completed) via the appropriate sub-sheets on the Environment Maintenance form and any differences between the two on the Differences sub-sheet.

The ini file **[JadeAppServer] DownloadVersion** is now only altered when the thinclient download area has been changed.

Thinclient Download Check

A new daily task is scheduled in order to check that :

- the thinclient download files have not changed
- that the ini file download version matches the calculated download version derived from the thinclient download area.

The timing of this schedule task is provided in the **Scheduling** sheet of the **System Global Options**.

This check is performed daily for all host environments (database and remote) for all environments defined as regime 3. However if a deployment is running or the JSA is not connected, the check is bypassed.

As the details are returned to JSM, they are evaluated and as well as updating each host environment, any variances will cause an alert to be generated.

Example of alerts (the content can also be combined or seen separately):

```
1. Thinclient download anomaly for TST32U340 of CNWMFJVB

   Thinclient download version in ini file does not match
   calculated download version for thinclient files.
   This could cause unexpected downloads on next deployment.

2. Thinclient download anomaly for TST32U340 of CNWMFJVB

   Thinclient download files for host environment TST32U340 of
   CNWMFJVB (database) differ from those defined for master
   environment:
   Files missing:
        i686-msoft-win32\bin/client.pem
        i686-msoft-win32\bin/freeimage.dll
        i686-msoft-win32/bin/zlib1.dll
```

Restore of Regime 2 Host Environment

While transitioning to regime 3, it may occur that a regime 2 environment is restored after the master environment has been set to regime 3.

In this case, JSA will detect the regime change and return the result to JSM. JSM evaluates the result and, if there is only one database host environment enabled or

there is only one SDS Primary enabled, will change the regime to 2 and purge the thinclient details held by JSM. If there are more than one database host environments or more than one SDS Primary for the master environment, then the regime is not changed.

In both cases an alert is generated advising of the regime conflict and what, if anything, has been done.

Example of alert:

```
1. Deployment regime conflict for host environment TST64U340 of
   CNWMFJVB

   Master environment TST64U340 regime 3 conflicts with host
   environment TST64U340 of CNWMFJVB.

   As only one database host environment or SDS Primary host
   environment is enabled for this master environment, the
   deployment regime has been reset to 2 and thinclient details
   purged

2. Deployment regime conflict for host environment TST64U340 of
   CNWMFJVB

   Master environment TST64U340 regime 3 conflicts with host
   environment TST64U340 of CNWMFJVB.

   As there is more than one database host environment enabled for
   this master environment, manual intervention is required to
   rectify this conflict.
```

Avoid Configuration Type Files Being Overwritten

In earlier release any specific files (such as ini or config files) that were NOT to be updated by a deployment needed to be explicitly coded for in JSA and cnwplugin.

As from hot fix 4.0.02.014 this is now catered for by additional verbs ExcludeFile<n> and RemoveFile<n> in section [JSMDeployment].

Refer to the ini file section [JSMDeployment] earlier.

Example:

An example is where a Jade Environment has .Net applications (running under one or more sub-directories in \server\c_misc\apps) and one or more requires a new configuration file named ‘appsettings.json’. The ini file settings below would ensure that the appsettings.json in the active directories are not overwritten accidentally by a deployment.

```
[JSMDeployment]
ExcludeFile1=d:\example\server\c_misc\apps\*,false,appsettings.json
RemoveFile1= d:\example\altos\ansi\server\x64-msoft-win64\misc\apps\*,false,appsettings.json
```

The ‘RemoveFile1’ would ensure that any files named ‘appsettings.json’ are removed from one level of sub-directories in d:\example\altos\ansi\server\x64-msoft-win64\misc\apps. For example if file d:\example\altos\ansi\server\x64-msoft-win64\misc\apps\mobileAppExample\appsettings.json existed it would be removed.

The “ExcludeFile1’ definition would ensure that any files named ‘appsettings.json’ found in the first level of sub-directories in d:\example\server\c_misc\apps are not propagated by JadeAdmin to any SDS and/or remote environments.

JadeAdmin Details

Automatic Deployment When Changed

As from the 4.1.01 release, when a base release of JADE is updated with anyone of the following, then the next deployment to an environment using that release (base or hotfixed) will automatically and unconditonally have those files plus .pdb files updated.

- jadeadmin.exe
- autodist.dll
- cnwplugin.dll

JadeAdmin and Single User App Server

When a JADE Environment has been defined as a Single User Application Server there is no requirement to maintain a JadeAdmin configuration file.

With the release of JSM hot fix 4.0.02.014 and its application to JSA, deployments to such environments will no longer execute JadeAdmin.

Ini File 'CreateConfiguration'

There are JADE Environments that are running as non-SDS capable and have no remote JADE fat-clients. These environments still require a JadeAdmin configuration file (as this is a requirement for JADECare managed environments and is checked by CardSchema [JadeAutoUpdate] EnableAutoUpdate=true).

It is desirable to avoid the maintenance of the JadeAdmin configuration file in such circumstances (and avoid sometimes lengthy JadeAdmin executions). A new ini file option has been provided as follows:

```
[JSMDeployment]
CreateConfiguration=<true|false>
```

The default for this value is false.

This ini file is only read when the environment's database role is SDS_RoleUndefined. When the database is a non-SDS environment and the ini file option is true then JadeAdmin will be run with argument 'createConfiguration' rather than 'updateConfiguration'. The 'createConfiguration' option does not require group membership and other details that can lead to lengthy delays in processing each file.

JadeAdmin Optional Script and Settings

The following features are available in JadeAdmin which can be used to enhance your deployment strategy:

Excluding Directories / Files

There are two options available only one of which may be used. . Both options can define directories / files and allows the use of wild-cards (*, ? and [<range>]). If both options are defined then the [plugin.excludefiles] will be the one used and [plugin.cnwplugin] is disregarded.

NOTE.

The exclusions defined are using regular expressions.

*If a whole directory is to be excluded then form should be <dir>/.**

If a wild card is to be inserted within a directory then form should be <dir partial>/./.*.json. If example d:/cnwtest/altos/server/ansi/x64-msoft-win64/misc/apps/.*/.*.json were defined, this would exclude the following files (should they exist):*

- *d:/cnwtest/altos/server/ansi/x64-msoft-win64/misc/apps/mobileservice/config.json*
- *d:/cnwtest/altos/server/ansi/x64-msoft-win64/misc/apps/samplerservice/sampler.json*

plugin.excludefiles

Defining the section [plugin.excludefiles] in the environment's configuration file (<env>.cfg) you can define directories and/or files to be excluded from JadeAdmin's inspection. That means that these will NOT be parsed or added to the manifest or deltamanifest and will NOT be propagated to any SDS Secondary or Remote JADE Environments. This could also improve the elapsed time of the JadeAdmin utility.

These entries can use the wildcard items * or ?

JSM automatically defines the following but these can be extended.

```
[plugin.excludefiles]
key1=<env_root>/server/c_bin63/.*
key2=<env_root>/server/c_system63/.*
key3=<env_root>/server/c_bin70/.*
key4=<env_root>/server/c_system70/.*
key5=<env_root>/server/c_bin71/.*
key6=<env_root>/server/c_bin_old/.*
key7=<env_root>/server/c_misc/app/.*/appsettings.json
key<n>=<<env_root>/<dir n> | <env_root>/<file n>
```

plugin.cnwplugin

This allows for the definition of a file that defines the exclusions.

```
[plugin.cnwplugin]
exclusions=<filename>
```

Include Copy From Altos

This feature is only available for regime 3 deployments.

By default, JadeAdmin only propagates the first level of the altos\....\misc directory to SDS Secondary and Remote Environments.

By defining one or more entries in the section [plugin.doCopyAltosFileFromStagingDir] in the environment's configuration file (<env>.cfg) this will ensure that the contents of that directory and any sub-directory will be folded down into the appropriate active directory (after deleting the contents of that directory).

```
[plugin.doCopyAltosFileFromStagingDir]
key1=<dir 1>
…..
key<n>=<dir n>
```

Example:

```
[plugin.doCopyAltosFileFromStagingDir]
key1=misc/apps
```

For an SDS Secondary, this will clear the directory <env_root>\server\c_misc\apps and will then copy down the contents (including

sub-directories) from <env_root>\altos\ \server\<character set>\x64-msoft-win64\misc\apps into <env_root>\server\c_misc\apps.

For a remote environment, this will clear the directory <env_root>\ c_misc\apps and will then copy down the contents (including sub-directories) from <env_root>\altos\ \server\<character set>\x64-msoft-win64\misc\apps into <env_root>\ c_misc\apps.

WARNING:

Within the JADECare 24 environment, any products registered as JMT environments should not include 'key<n>=misc/htdocs' as the 'htdocs' directory is maintained via the jmt_htdocs.rar and _JADE_Deletion_manifest.txt via the deployment module and, for remotes, using the <env>_prerestart.ps1 or cmd script.

Pre Restart Script

The pre-restart hook is restricted to:

- SDS Secondary environments and
- Remote environments and
- environments using deployment regime 3 or later and
- file <env_root>_preRestart.cmd or <env_root>_preRestart..ps1) exists within the parent's \c_misc\cardinal directory.

When JadeAdmin executes on the SDS Secondary environment then just before the service restart phase, it will execute the file <env_root>\server\c_misc\cardinal\<env_root>_preRestart.cmd or <env_root>\server\c_misc\cardinal\<env_root>_preRestart.ps1 if the file is present.

When JadeAdmin executes on the remote environment then just before the service restart phase, it will execute the file <env_root>\c_misc\cardinal\<env_root>_preRestart.cmd or <env_root>\c_misc\cardinal\<env_root>_preRestart.ps1 if it is present.

Note that any standard output from this script is logged in the JadeAdmin.log.

Setting up for Auto-Updating Deployments

Overview

When deploying to JADE 6.1 systems, JSM 3.1 can use new features available in 6.1 to achieve the following:

- Automatic update of app servers and fat clients
- Automatic update of SDS secondaries and RPS nodes

The JADE components required to achieve this are (initially) released separately from the standard JADE release consistency, and should be requested from the JADE Care plant. The components are:

- jadeadmin.exe
- autodist.dll
- cnwplugin.dll

Setup for Automatic Fat Client Update

1. The master environment object in JSM must have deployment regime set to '2'. This requires that the managed environment be on JADE 6.1.07 or later and using CardSchema version 3.1.01 or later, and that JSA on the managed host be at level 3.1 or later.
2. Client initialization files must have the **InitializationHandlerLibrary** statement changed from **mismatch** to **jomsupp** (i.e. the JADE default):

```
[JadeClient]
InitializationHandlerLibrary=jomsupp
```

3. The initialization file for each managed database server and fat client node must have the following entries added:

```
[JadeAutoUpdate]
baseDirectory=<drive>:\<env id>
EnableAutoUpdate=true
CurrentConfiguration=<env id>.cfg
helperLibrary=autodist
Distribution=i686-msoft-win32
clientUpdateIfMissing=true
autoRestart=false
autoShutDown=false
commandLine=true
```

4. Ensure that the following statement is in the ini file (Linux example only):

```
[JadeLog]
LogDirectory=/opt/test31/logs
```

5. The three components listed in the overview above must be obtained (must be applicable to the JADE version in use), and copied to each managed node's binaries directory.
6. Generate the initial JadeAdmin configuration file on the database server by modifying then running the supplied RunJadeAdmin script. An example command line for running JadeAdmin to create a new configuration file is:

```
F:\test3161\server\c_bin\jadeadmin.exe
ini=F:\test3161\server\c_bin\test3161.ini
baseDirectory=f:\test3161
config=test3161.cfg createConfiguration JADECare
```

Setup for SDS Secondary & RPS Node Update

1. Ensure that the SDS secondary or RPS node is using the same version of binaries as the primary.
2. Perform setup as for fat client (steps 1 to 5 above), on the primary server.
3. Update the secondary's ini file and binaries (steps 2 and 3 above)
4. Copy the generated config file from the base directory on the primary system to the base directory on the secondary system.
5. If the SDS secondary's host environment name is not the same as the primary's, rename the secondary's config file to match, e.g. if the secondary's host environment name is TEST3161B then the config file should be renamed 'test3161b.cfg'.
6. Ini files for RPS nodes must contain a number of options. Example options from an ini file are shown below. Options shown in bold type are required values. JADE automatically sets up some of these values.

```
[JadeRps]
AutoStartDataPump=true
RPSConnectionStr=DSN=TEST3161R_RPS;Database=TEST3161R_RPS
RDBName=TEST3161R_RPS
RDBLogin=false
RDBPath=\\cnwrwm1c\share60\sql
AutoScriptPath=\\cnwrwm1c\share60\sql
AutoRunReorgDropScript=true

[JadeRpsManager]
ExtractDir=\\cnwrwm1c/share60/sql/
ExtractOverwrite=true
ScriptPath=D:/Share60/SQL/
```

7. For SDS environments that include RPS nodes that use Working Sets, the following ini file options may be required on the **primary** (values shown are examples only):

```
[JadeRps]
AutoExtractOnPrimary1=TestSchema,TEST3161R,\\cnwrwm1c\shar
e60\sql,\\cnwrwm1c\share60\sql,TEST3161R_RPS

[JadeRpsManager]
RDBName=TEST3161R_RPS
Schema=TestSchema
Mapping=TEST3161R
ExtractDir=\\cnwrwm1c\Share60/SQL/
ExtractOverwrite=true
CreateDatabaseDir=F:/TEST3161R/server/c_system/
CreateOverwrite=true
CreateCopyMode=Online
ScriptPath=D:/Share60/SQL/
```

   Refer to the applicable JADE product documentation for more information about RPS ini file options.

Summary of Deployment Process for SDS Systems

JADE 6.1 systems that are set up with environment deployment regime=2 in JSM can have SDS secondaries and RPS nodes automatically updated during deployments. The following outline describes how the SDS secondary update process is controlled. Note that the order of events is different depending on whether the primary deployment was conducted on-line or off-line, and depending on whether any schema files and reorganisations occurred during pre-installation or post-installation script executions. So this is a somewhat generalised description.

1. JSA on the SDS primary marks the start of the deployment via an sdsAuditStopTracking entry into the audit trail, with a reason code of 100. When this is processed on the secondary, if a pre-installation backup was requested for the secondary, then this backup is initiated, otherwise go to step 2.
2. Once any backup is completed, the secondary waits until both of the following conditions are met:

   - Any deploy_hold.htm file is removed
   - Deployment is completed on the primary

3. Once the above conditions are satisfied, JSA automatically resumes tracking (restarts data pump for RPS nodes) and the secondary node processes any reorgs. For RPS nodes, if a reorganisation is encountered that requires manual

intervention (e.g. to load table alter scripts and extract and reload data), tracking stops (RPS data pump stops) and JSM alerts and displays a desktop message.

Once the RDB has been manually updated, the JSM user selects the 'Continue' option on the deployment control form, and JSM restarts the RPS data pump.

Note : During deployments, adminstrators must not manually stop or start SDS tracking (using the JadeSDSAdmin application), or start or stop the data pump (using RPS Manager), otherwise JSA will lose synchronisation with the database state.

For RPS nodes where JADE will automatically execute a table alter script, JSA silently handles any tracking stoppages. Note that this requires ini file option **[JadeRps] AutoStartDataPump=true**, and that the RPS node is configured to sign-on and authenticate with the target RDB server without user intervention.

4 After reorganisations have been processed by the secondary, tracking will stop for a version mismatch audit entry. This is inserted into the audit trail by the primary when the JadeAdmin utility is executed to update the <env>.cfg file. This file contains the md5 hashes of all binaries and other files that need to be transferred to the secondary. The operator is prompted to approve the secondary system shut down for transition and binary update. The installation process optionally holds at this point until the operator approves the system shut down.

5. JadeAdmin is notified by JSA that the system has been shut down, and updates the binaries and other files on the secondary, copying from the primary system's directories.

6. JadeAdmin notifies JSA that system restart is to be initiated. JSA restarts the secondary system.

7. JSA monitors the secondary after restart. Additional reorganisations may occur during this period, depending on the configuration and content of the deployment. When the secondary database processes the end-of-deployment tracking stop (reason=109, inserted by the primary via sdsAuditStopTracking) JSA then launches KCCMgr if required, then automatically resumes tracking (or restarts the data pump) and initiates any required post-installation backup.

8. Note that SDS secondaries and RPS nodes that run user-defined server applications should have the ServerApplication statements in the ini file defined in the [Karma] section rather than in the [JadeServer] section, so that KCCMgr can launch them. Otherwise these applications may not be relaunched after certain types of deployments to SDS secondaries where the primary deployment was performed off-line.

Recovery from SDS Primary Deployment Failures

Online Deployments

If deployment to a primary fails during the **online** phase of installation, (e.g. schema files have load errors) then the deployment abort at the primary will abort any reorg and unversion all schemas. Any running SDS secondary will generally (but not always) be waiting at a tracking halt (reason=100 : Start of deployment, which is audited by the primary after the primary has taken the pre-installation backup but before loading any schema files). This status will be apparent from

review of the SDS or RPS log, or on the HostEnvironmentMaint screen's SDS sheet for the secondary.

If deployment to an SDS primary fails during the **offline** phase of an online deployment (i.e. during or after transition) then the deployment abort at the primary will restore the primary from the pre-installation backup and replay journals up to the point of transition. The JSM operator must then restart the primary service. The secondary will be waiting at the start of deployment tracking halt, or will be waiting for the primary deployment to complete, as for failures during the on-line phase of the installation.

Example 1

Scenario :
Online deployment suspended in JSA waiting for scheduled transition time but jadrap crashed after online reorg and transition. Refer to PAR 6698.

In this case the jadrap was down and the deployment state was unaware of this as waiting for scheduled time.

JSM's deployment control form could still be used to 'Abort Installation' which would execute the necessary commands to 'abort the reorg' and 'unversion all schemas' and leave the database in a usable state and the JARI in a 'Pending Restart' state.

Restarting an Installation

If the deployment is to be **restarted** and the installation was aborted before schema transition occurred, then JSA will not repeat the pre-installation backup (and will not re-audit the start of deployment journal entry, so the operator should **not** resume secondary tracking or restart the RPS data pump. Each secondary will remain waiting until deployment to the primary is completed, at which point JSA will automatically resume tracking. So if the primary deployment fails, it can be aborted and restarted without requiring special steps to handle the secondaries.

Abandoning a Deployment

If the deployment is to be **abandoned** (i.e. not restarted) then this situation currently requires the following manual actions from the JSM operator::

(a) Using the JSM deployment control form, abort the primary installation. This will also automatically abort all secondary installations.

(b Kill the deployment using the deployment control form.

(c) From the JSM GUI use the menu option **Deployment | Environment Reset** and specify the name of the environment in the text box provided. This resets flags in JSM *and* JSA. Refer to section below describing **Environment Reset.**

(d) Restart the primary.

(e) For each secondary, use the SDS Admin application to resume tracking if necessary (or for RPS nodes use the RPS Manager to restart the data pump). If the primary deployment failed after transition, you may need to do this

twice : once to resume past the start-of-deployment halt, and again to resume past a reorg halt.

Environment Reset

This is provided as a menu option under the main menu option of **Deployment**. It replaces the previous implementation of a JadeScript::environmentReset that required usage of the IDE.

The environment reset performs the following functions:

- Resets the JSM flags for the master environment and host environments concerned.
- Resets JSA flags for the master environment and host environments concerned.
- If environment contains a SDS Primary provides the facility to allow the JARI to be marked as Completed. Otherwise the default is to mark the JARI as Requires Scheduling.
- If Tracker implemented, generates a Tracker call to the JSM Administrator (FMS Assignee for JSM host environment) for follow up.
- If Tracker not implemented, an alert is created advising that the environment has been reset to ensure follow up.

Recovery from SDS Secondary Deployment Failures

Aborting SDS Secondary Installations

In the event that installation fails at the secondary, JSM allows you to use the **Abort Installation** option on the Deployment Control Form to reset the installation state for the selected host environment (SDS secondary or RPS node). Note that this action does not change the current state of the target host environment, and that you should **not** use the SDSAdmin or RPSManager tools to resume tracking or restart the data pump, except as described in **Manual Completion Procedure** below.

Unlike SDS primaries and non-SDS systems, the installation abort process for secondaries does not perform a restore operation from the pre-installation backup. Once secondary installation proceeds past the point where the primary deployment completed, there is no going back.

After aborting a secondary installation, you can **retry** the installation by using the Start Installation menu item on the Deployment Control Form. Normally, JSM/JSA will evaluate the state of the secondary database depending on the current tracking status and other indicators, and resume after resynchronizing with the target environment.

Recovery Procedure After RPS Autoload Failure.

As from release 3.9.02, the deployment engine detects if there has been a failure during the RPS Reorg Halt. This is the phase where the datapump will:

- Generate an alter script
- Extract requisite files to load new tables into the SQL database
- Execute the alter script which will can:
    - Change existing tables to be prefixed ‘Jade_Dropped_’

- Add new tables and procedures
- Load the data into these new tables

During any phase of the above, there can be a failure for a variety of reasons. Such failure will cause an error file to be created named **rps_<uniqueid>.err** and this will be placed into the directory defined by **[JadeRps] DatapumpErrorDirectory=<directory>**. The deployment engine will set this by default to be the same as the directory for [CardLog] LogPath.

Finding this .err file during the RPS Reorg Halt will cause the installation to go to error state and will display as much of the content of the file as possible. It may be that to fully understand the root cause of the issue that you will need to:

- Review the contents of the rps_<uniqueid>.err
- Review the jommsg.log for the time period when the .err file was created
- Reviw the JadeRpsDatapump.log (if it exists)

To recover from this error:

1. Request 'Abort Installation' for the RPS node entity on the Deploy Control form
2. Resolve the issue identified in the .err file. This may require manually extracting scripts and class details and then editing the script before executing it via the RPS Manager
3. Restart the datapump as required until you have incurred Tracking Halt 102 or 109..
4. Once you have identified that the RPS node has a tracking halt of 102 or 109, then restart the installation for the RPS node by clicking on the menu item 'Start Installation' and this will allow the RPS node installation to go to successful completion.

Manual Completion Procedure

In the event that deployment to an SDS secondary encounters errors which you are unable to bypass, you can manually complete the installation by following the steps listed below. This should avoid any requirement to restore the secondary from backup, or to reclone the secondary.

This procedure assumes that the primary has completed its installation, and that one or more secondaries have installation problems that cannot be easily resolved.

1. Use the JSM Development Environment to run **JadeScript::environmentReset**. This will reset all flags related to the deployment and leave the JARI in state 'Requires Rescheduling'. You should now cancel this JARI.
2. In the target environment's base directory (<driveLetter>:\<env name>), remove the **jadeadmin.lck** file, if present. If the jadeadmin.exe or jadeadmin2.exe program is running, kill it.
3. Copy the <env name>.cfg file from the SDS primary's base directory to the secondary's base directory. Rename the file if the host environment name is different from the primary's host environment name.
4. Copy the latest **xfer** directory from the primary to the secondary's **xfer** directory. This directory will have a name like **000007.<hex string>** where the 000007 represents the current sequence number as defined in the **seq** statement in the JadeAdmin config file (<env name>.cfg).

5. Copy bin, lib and system files for the appropriate database platform from the **xfer** directories to the active environment bin, lib and system directories.
6. If this is an SDS Secondary Relational database (RPS node), then ensure that the [JadeServer] ServerApplication1=CardSchema,KCCMgr entry has been defined as this maybe temporarily deleted by the deployment process and needs to be redefined when employing manual recovery procedures.
7. Start the secondary system if it is not running.
8. Start the **SDSAdmin** application for the secondary system, and the **RPSManager** for RPS nodes. You must now pass through one or more tracking halts until the SDSAdmin tool shows that your secondary is synchronised with the primary. Each time after restarting tracking (or for RPS nodes, restarting the data pump), you must review the sds log (or rps log) to determine the reason for the latest tracking stop. As you will see from the diagrams in Appendix A, the following broad sequence of tracking halt reason codes occurs. There may be more or fewer of these, in varying orders, depending on the type of deployment.

Start of deployment	100
End JADE version upgrade	101
End user schema load phase	102
RPS admin halt	4
Schema transition	5
Version check	6
End of deployment	109

   Once you resume tracking past the 109 halt, the database should remain tracking and synchronised.
9. Back up your secondary database, if required.

Deployment to RPS

When the installation phase for a RPS node starts up the following checks/updates are performed:

- If the **[JadeRps] AutoRestartDataPump** is not true, the installation will go into error state and request that the ini file is fixed up and the RPS node recycled.
- If the ini file setting **[JadeRps] JSMDeploymentAlterScriptOverride** is not defined it will be inserted into the ini file with a value of true.
- When the ini file setting **[JadeRps] JSMDeploymentAlterScriptOverride** is false, then the values and context of the alter script related ini options are logged, these settings being:
    - [JadeRps] AutoRunReorgDropScript
    - [JadeRps] AutoBulkLoadHistoricalTables
    - [JadeRps] DropHistoricalTableOnAddExisting
- When the ini file setting **[JadeRps] JSMDeploymentAlterScriptOverride** is true, then the values of the following alter script related options are unconditionally set to true:
    - [JadeRps] AutoRunReorgDropScript
    - [JadeRps] AutoBulkLoadHistoricalTables
    - [JadeRps] DropHistoricalTableOnAddExisting
- The **[JadeRps] DatapumpErrorFileDirectory** setting is unconditionally set to the CardLog log path and any files that match file mask rps_*.err are renamed to contain suffix .log, so that they are subject to housekeeping.

Deployment of Updates to JSA

Updates to JSAs can be manually deployed, or deployment to Windows JSAs can be semi-automated using the JSM Deploy Process. In order to partly-automate the deployment of updates to JSAs ('karma'), you must supply a customised **install_karma.bat** file as part of the deployment. JSM knows it is a JSA deploy and will unzip the files and initiate the **install_karma.bat** file to perform that actual upgrade.

Note that JSM 3.3 can only deploy JSA updates to JSAs that are 3.2 level or higher

Check that ALL **ccc**karmas (where **ccc** is your customer id) have the correct version files before beginning

Create empty directories on your PC to hold the c_bin, c_misc and c_system directories. You will always need a misc directory for the install_karma.bat script.

Copy appropriate binary and system files from the JSM release to the folders you created in the previous step.

Copy **install_karma.bat** to the misc directory that you created above.

Go to the 'System, Method and CLS file loads' section of the script and add/update installs of the appropriate JossSchema, CardSchema, JadeSchema, CLS loads and method patches. Make sure that you remove the 'rem' from the beginning of the lines that you update.

Build a Jari for ccckarma and distribute the files without specifying an installation start-time.

Once distribution is complete, use the Start Installation menu on the deployment control form. JSA will launch the script. Output is directed to the **logs** directory in **install_karma.log**.

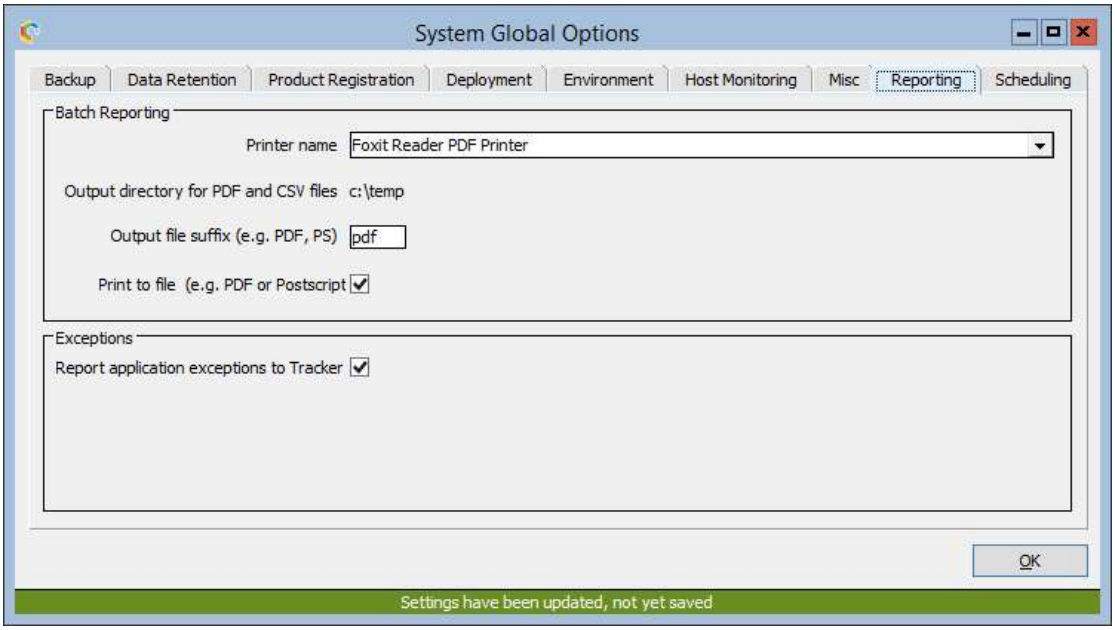
Performance and Utilisation Reporting

Overview

JSM provides comprehensive ad hoc and batch reporting facilities for host and application performance measurements. Reports can be run interactively, or can be scheduled using report sets for end of month or end of week style reporting.

Set Up

System global parameters for reporting are set up on the System Options form's Reporting sheet.



NOTE: When directing output to a PDF file, then the printer name selected must be that of a PDF converter driver.

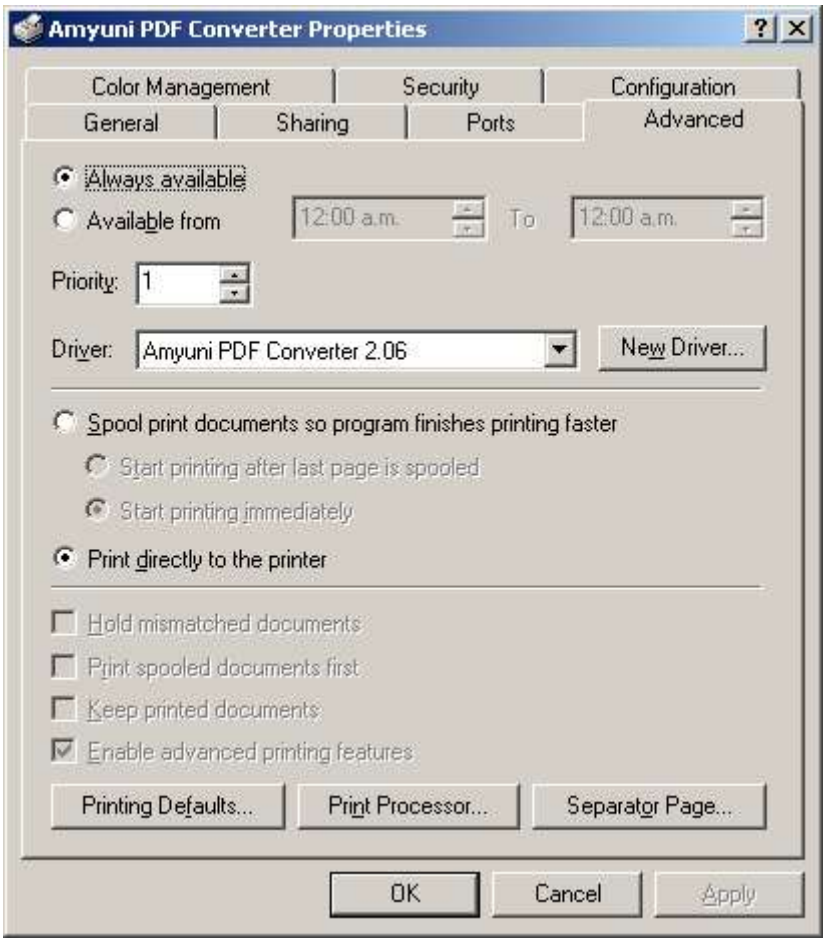
For interactive (ad hoc) reporting, run reports using the **Reports | Initiation** menu item in JSM.

For batch reporting, define Reports, Report Sets and Report Groups from the **Reports** menu in JSM.

Graphs displayed on the desk top.

Most graph type reports can be closed down at once by holding the **Shift** key when clicking on the Close control on any one of the forms concerned.

If it is desired to direct batch report output to a PDF printer, then the recommended PDF driver is Amyuni PDF Converter 2.06 or later (2.10 currently recommended). When configuring this driver, ensure that the option to print directly to the printer is set, as shown below. If this is not done, hidden desktop messages will be generated, requesting output file names.



Host Performance

Application Resource Utilisation

Transaction Performance

Summary Logs

There are a variety of activities that are logged by JSM, notably:

- Deployments (3 distinct phases logged separately – Assembly, Distribution, Installation)
- Backup
- Restore
- Miscellaneous (eg. Global Parameter distribution)

This information is retained for a period specified in the System Options form (sheet Data Retention, item 'Retention days for miscellaneous logs (deployment, backup etc)').

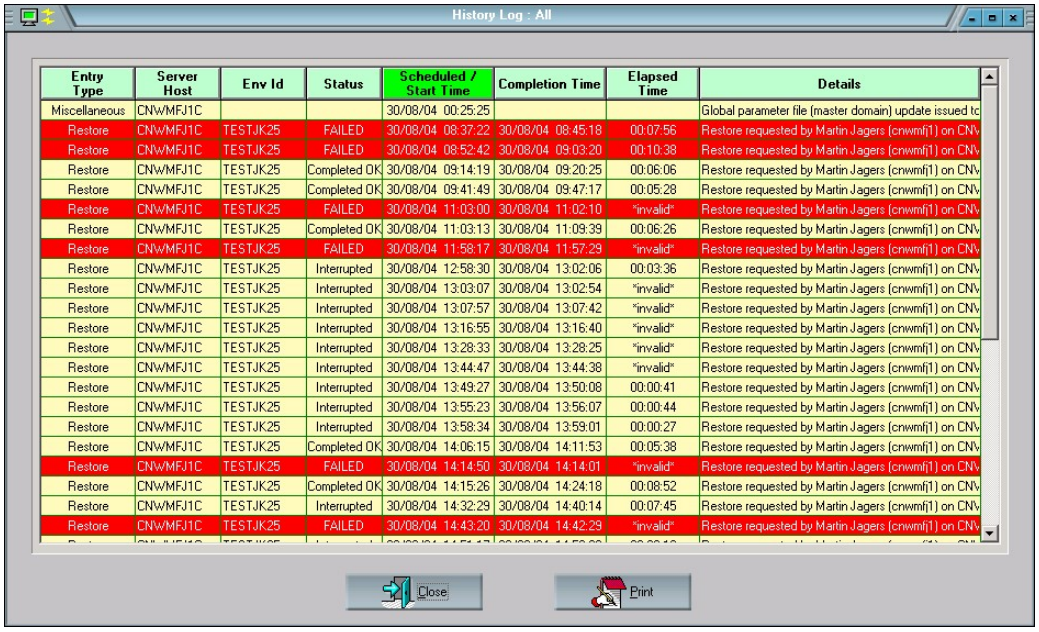
Selecting Logs

This information may be selected by a variety of parameters and either displayed or printed via the following screen.

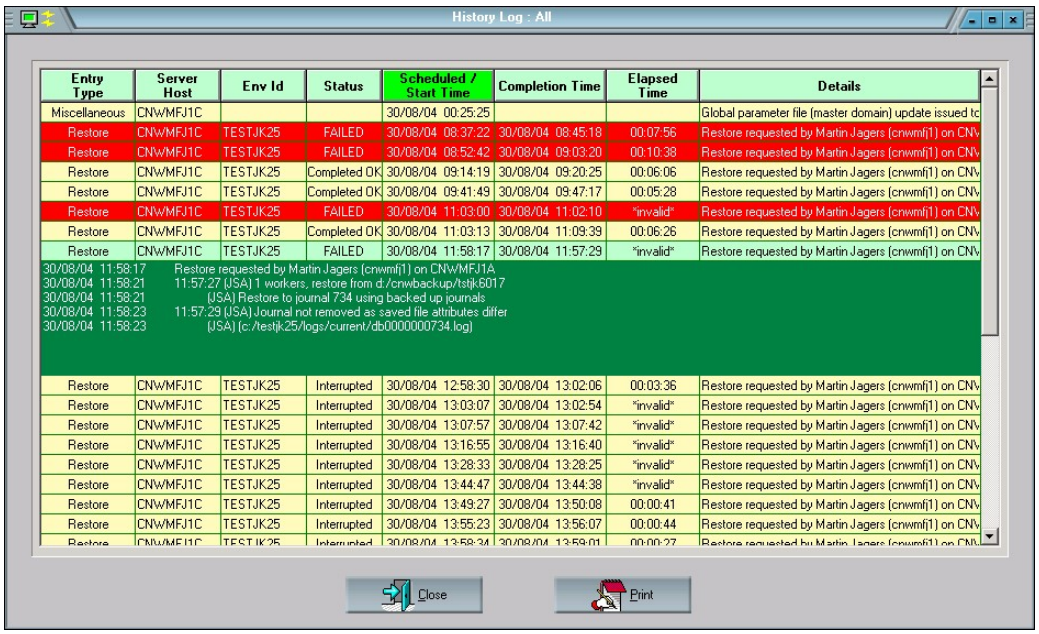
The server host and environment selections are both optional, if neither are specified than all logs meeting the remaining criteria will be selected.

Display Logs

The following is an example of the form displayed arising from the selection criteria defined above, items shown in red indicate that the activity concerned had errors:



Clicking on a row depicting an activity will cause a summary of the log details recorded by JSM to be displayed immediately below the row clicked:



Printing Logs

If the print option is clicked on the selection form above (Select Logs), then a print out of each log accompanied by all the JSM recorded details for that log are provided in hard-copy.

Global XCF Parameters

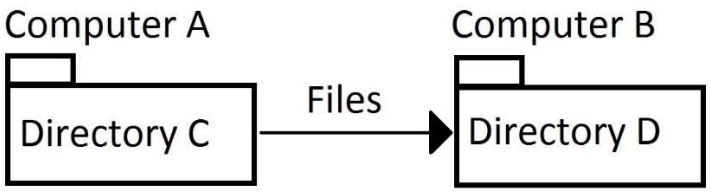
This facility allows directories of files from one host to be automatically forwarded to another host via CNWCommander.

Replication will start after issue of (ad-hoc or scheduled) Global parameter distribution to the source host.

XCF Item

A unidirectional file copy specification from host A to host B is established when an operator creates an XCF Item (see the "Global XCF Item Maintenance" section below). The XCF Item defines the source host and directory and the destination host and directory. An XCF Item may also define other aspects of the copy such as whether files in subdirectories should be copied as well.

Fig 1 XCF Item Supports Unidirectional File Copy



XCF Item Pair and Pair Switching

XCF Items may be defined in pairs where each member of the pair dictates a file copy direction *usually* opposite to its counterpart. I.e. One XCF Item specifies file copy from host A to host B and its counterpart specifies file copy from host B to host A.

Typically, an XCF Item Pair is defined to enable easy file copy direction reversal (switching) as part of maintaining a Disaster Recovery (DR) configuration. At a given time, only one XCF Item of a pair is active (the filled-in arrow in the diagrams). Switching an XCF Item Pair changes which XCF Item in the pair is current, and hence switches the direction of file copy.

In an XCF Item Pair, the XCF Item that is the *normally* current is referred to as the ***Primary*** XCF Item. The XCF Item that is *not* normally current is referred to as the ***Secondary*** XCF Item.

Fig 2 XCF Item Pair, XCF Item 1 (host A -> B) is current and file Copy is from C to D

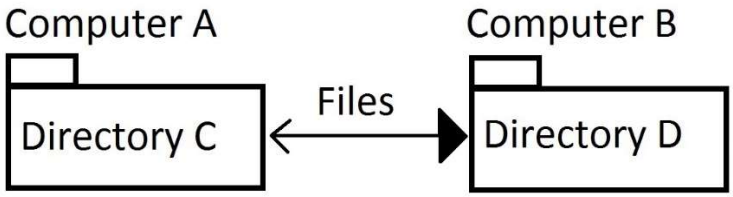


Fig 3 Source and Destination directories can be different on a given host in an XCF Pair

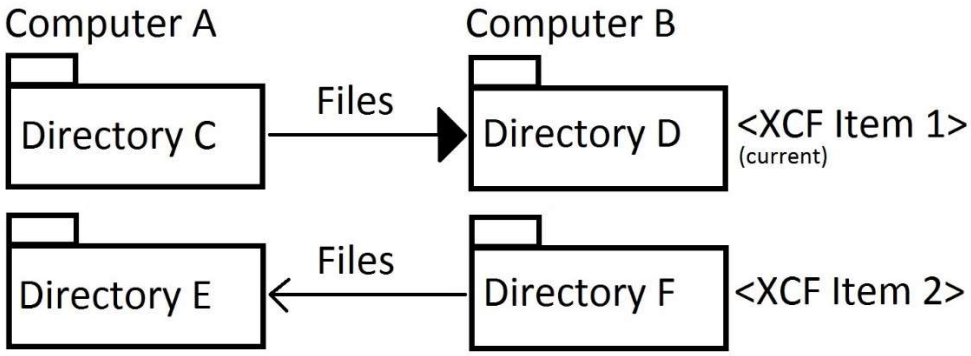
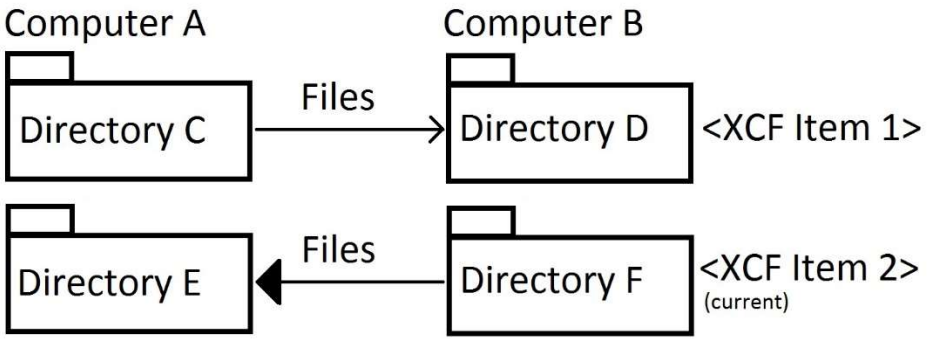


Fig 4 XCF Item Pair post-switch, XCF Item 2 is current and file Copy is from F to E



XCF Group and Group Switching

In some cases. multiple XCF Item pairs exist that relate to each other such that they should not be switched independently. A mechanism is provided for assigning two or more XCF Item Pairs to a group such that when a switch is applied to the group, all XCF Item Pairs in the group are switched.

Fig 5 Two XCF Item Pairs in an XCF Group

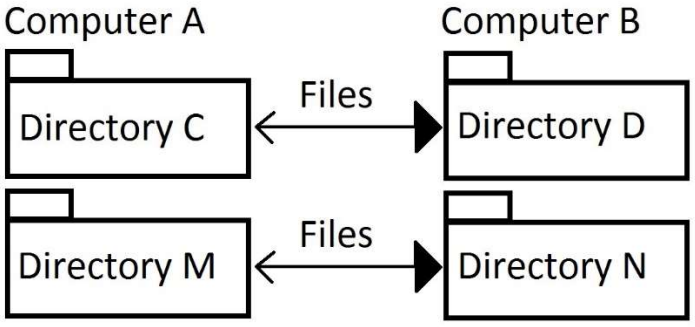
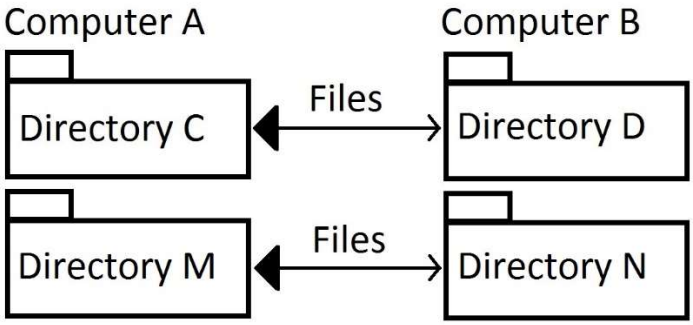


Fig 5 Two XCF Item Pairs in an XCF Group, Post Group Switch



At a given time in a group, it may be desirable to have file copies in both directions to support independent functions (in other words, there is no requirement that the current Item in each Pair copy files in the same direction). For example, in the case that a web application runs on the DR host, it may be desirable to maintain files on the DR host as well.
It is a requirement however, that either all Primary XCF Items in a group are current, or all Secondary XCF Items in a group are current (to preserver the Primary or Secondary configuration).

Fig 6 Two XCF Item Pairs in a Group, Supporting Independent Functions

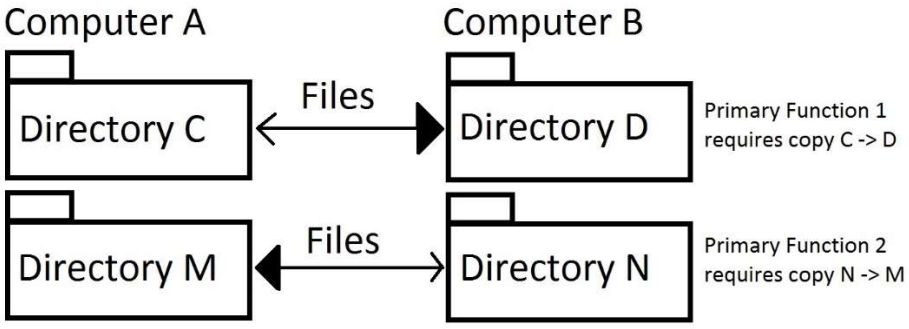
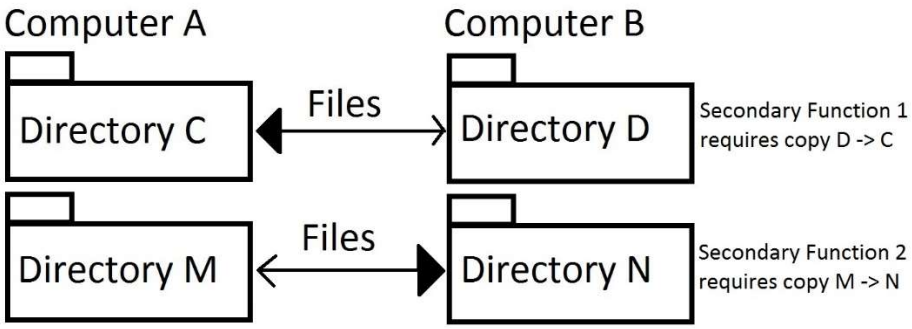
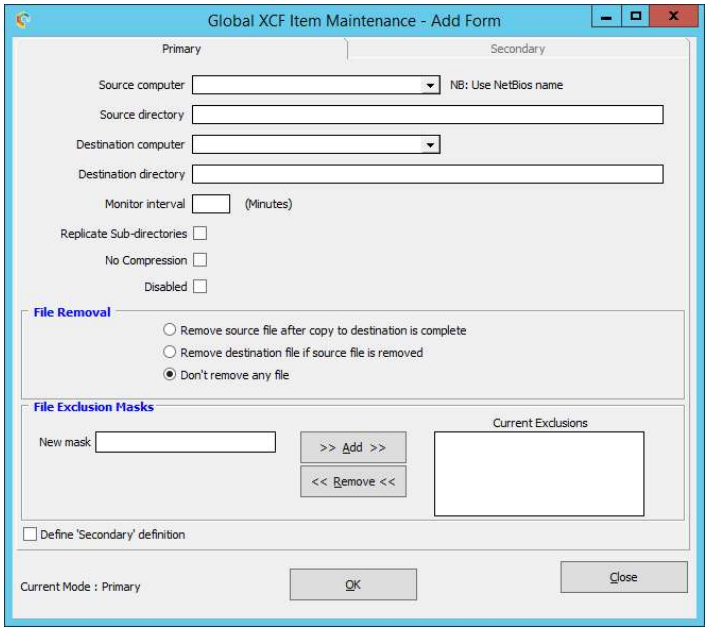


Fig 7 Two XCF Item Pairs in a Group, Supporting Independent Functions Post-Switch



Global XCF Item Maintenance



When adding a new item where Monitor Interval is specified or when updating an existing item and Monitor Interval is being changed, the user is issued with a pop-up dialog advising that overheads of using this could potentially be large. The user needs to OK this dialog for the update to proceed.

Note: The preferred option is to use a Monitor Interval of zero (0). This means that any file is creation or update in the nominated source directory will cause a notification to CNWCommander and cause a copy of the file to the specified destination directory.

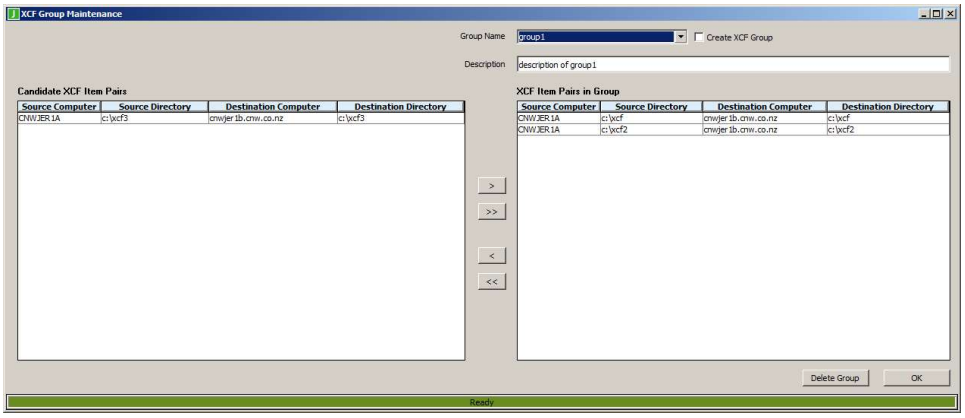
Whole directories can be refreshed by use of the Monitor Interval but the overhead of this can be large unless the directory contains few files and the monitor interval is high.

The user can define a secondary item by clicking the “Define ‘Secondary’ definition” check box while creating a new item or updating an existing item.

- The Secondary will have identical values to those on the Primary sheet except source and destination hosts will be swapped. The user can change/enter these values as they wish

The ‘disable’ flag allows the item to be defined within JSM but prevents it from being exported to servers for use by Commander. The setting is applied to the Item partner if the XCF Item exists in an XCF Item Pair.

Global XCF Group Maintenance

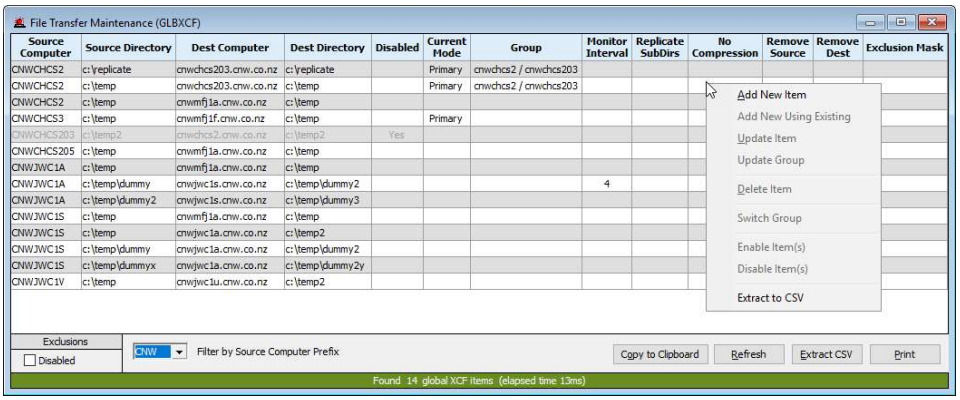


The XCF Group Maintenance form provides the ability to create (and delete) named Groups that have XCF Item Pairs such that when a Group Switch is performed, all Items Pairs switch which Item in the Pair is current.

In the picture shown above, the user is adding a new XCF Item Pair to an XCF Item Group that already has two XCF Items Pairs.

The current Item of each pair is displayed.

Global XCF Summary



The above form shows all the Global XCF parameters defined to-date. Disabled entries will be shown in grey colour and only visible if 'Disabled' exclusion is not checked.

A combo box is provided which can be used to display only those GlbXcf items where the first 3 characters of source computer match.

The form permits multiple user access but checks update version to ensure changes are not inadvertently lost.

Right clicking within this form provides a menu selection to:

Function	Description
Add New Item	Create a new XCF entity. User can define a secondary item by clicking "Define 'Secondary' definition" check box.

Add New Using Existing	Create a new XCF entity with most items prefilled from the selected item (including secondary item fields if the selected item contains a secondary item).
Update Item	Opens the XCF Item Maintenance form Update the selected XCF Item (and its Secondary).
Add Item To Group	Opens the Group Maintenance form. Add an XCF Item Pair to a (new or existing) Group
Update Group	Opens the Group Maintenance form. Remove XCF Item Pairs from an XCF Group. Delete the Group.
Delete Item	delete a selected XCF entity If the selected item is a Primary or Secondary, this menu item will become 'Delete Primary & Secondary Items' and it will delete both items.
Delete Secondary Item Only	This menu option will only get shown if the selected item is a member of an XCF Item Pair, and it will delete the Secondary item only. If the secondary item is current, then deleting the Secondary will make the Primary current.
Switch Item	Switch between Primary and Secondary XCF Items in a Pair.
Switch Group	Switch all XCF Item Pairs in a Group.
Enable Item(s)	Mark all selected items as Enabled.
Disable Item(s)	Mark all selected items as Disabled.
Extract to CSV	Extract to CSV.

XCF Item Pair and XCF Group Rules

All XC Item Pairs in a Group belong to a Group configuration that is either Primary or Secondary; meaning that all Pairs in the Group have either Primary or Secondary Item current respectively.

When adding XCF Item Pairs to an XCF Group, all Pairs must have the same Mode (either Primary or Secondary) and that mode must match that of the Group.

When the Seconday Item of a Group Pair is deleted, the Primary Item is made current and removed from the Group.

OS Patch Maintenance

This optional facility allows the Security Administrator visibility to installed operating system patches for each Host registered in JSM. The facility will only operate when the scheduled entry is selected and a runtime defined which will cause JSM to run a daily scheduled task called 'OS Patch Reader' (application **JossOSPatchMgr**).

The **JossOSPatchMgr** application reads files transferred from each Host using tools provided by JADECare 24. The files are located in a repository local to JSM and is identified by the following initialisation file option:

[JadeSystemsManager]

OSPatchHistoryDirectory=<directory>

The following two sub-directories are expected to be found under the directory named above :

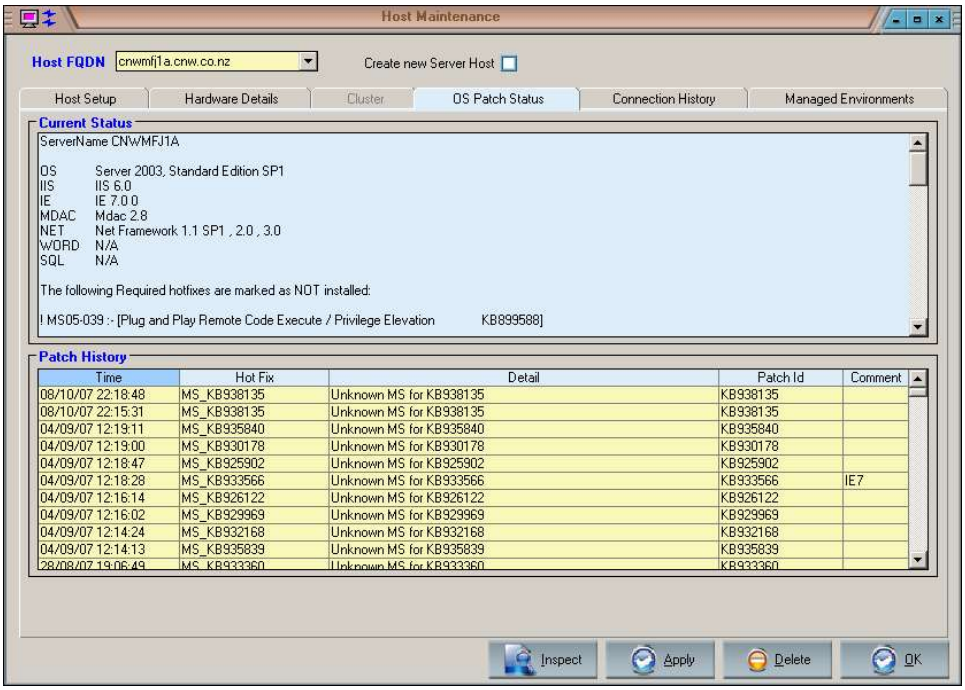
- \server_config\
- \server_hotfixcheck

If any of these directories do not exist, alerts will be raised on each JossOSPatchMgr run and the run will be aborted.

The JossOSPatchMgr application can be run as required via the 'File | Initiate OS Patch Reader Run' menu item.

Viewing Hot Fixes Applied:

The OS Patch Status details can be viewed via the Host Maintenance form on sheert **OS Patch Status**.



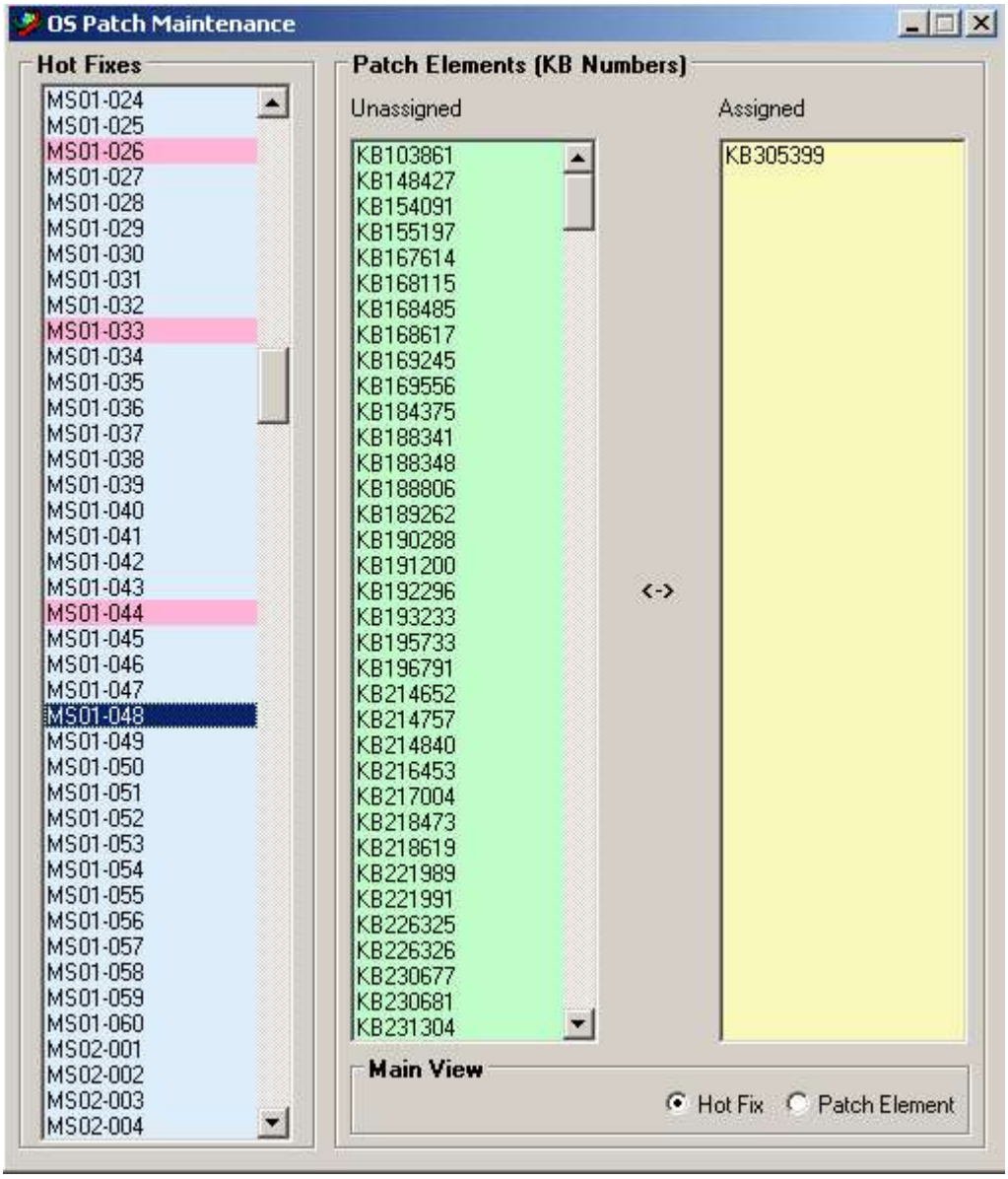
The upper-half of the form represents a summary derived from \server_hotfixcheck\<serverhost>_Updatestatus.txt.

The lower-half represents Hot Fixes loaded on this server to date.

Note that the pink colour denotes 'critical' Hot Fixes. Bubble help exists on each entry indicating if more KB entries (Knowledge Base Articles) are associated with the Hot Fix.

Maintaining Hot Fix and Knowledge Base Articles:

The Security Administrator has the ability to assign/unassign KB articles to/from specific hot fixes via the OS Patch Maintenance form. This can be displayed by clicking the 'Setup | OS Patch Maintenance' menu item from within JSM.

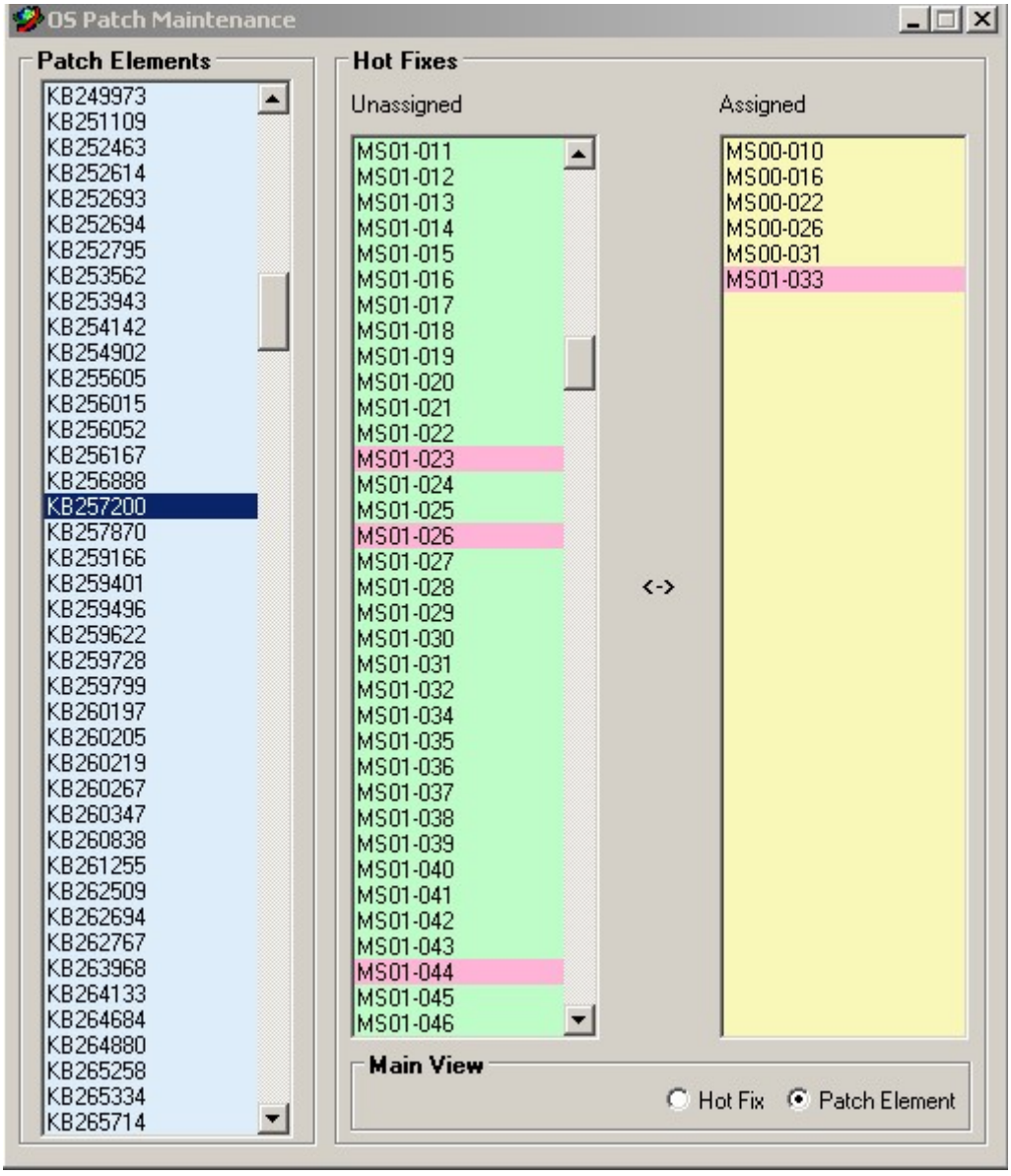


This form can used to view Microsoft Patches from two perspectives:

- by Hot Fix number
- by Knowledge Base Article

When viewing from perspective 'Hot Fix', you can assign KB numbers to the selected hot fix by double clicking an entry in the 'Unassigned' list box. To unassign a KB number double click the desired entry in the 'Assigned' list box.

When viewing from perspective ‘Patch Element’ …

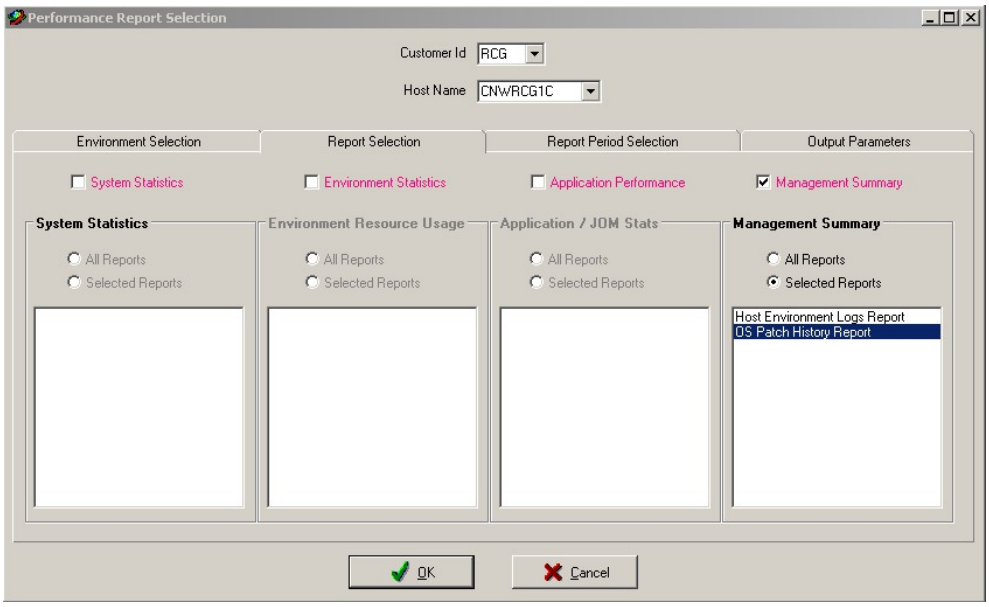


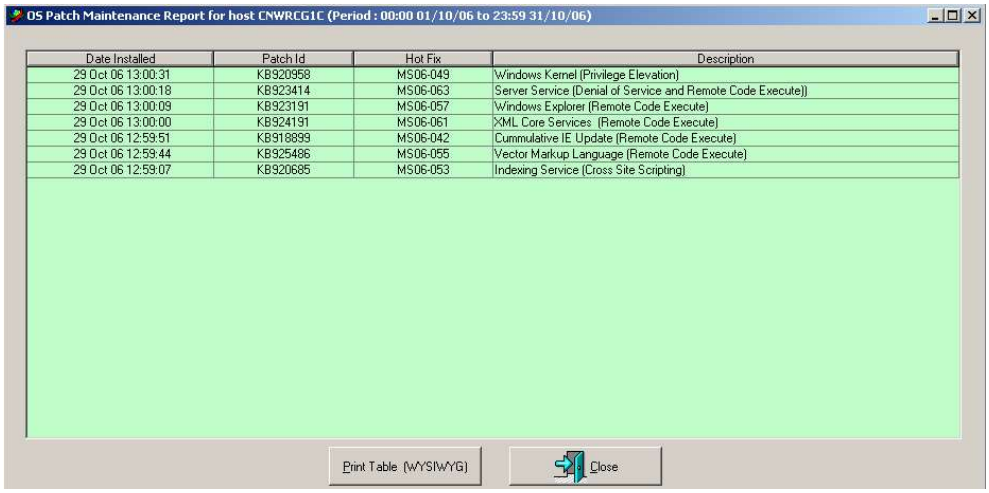
This can be useful if you want to see at glance what hot fixes are applicable to the selected knowledge base article. The double clicking behaviour of both the ‘Unassigned’ and ‘Assigned’ list boxes is the same as the ‘Hot Fix’ view mentioned above.

This form is resizable and it’s position and size will be retained after each use.

Reporting on Hot Fixes:

A new report has been created for displaying/printing ServerHost Hot Fix load information during the criteria specified.





In the example above the parameters provided requested all hot fix information for the period of October against ServerHost CNWRCG1C.

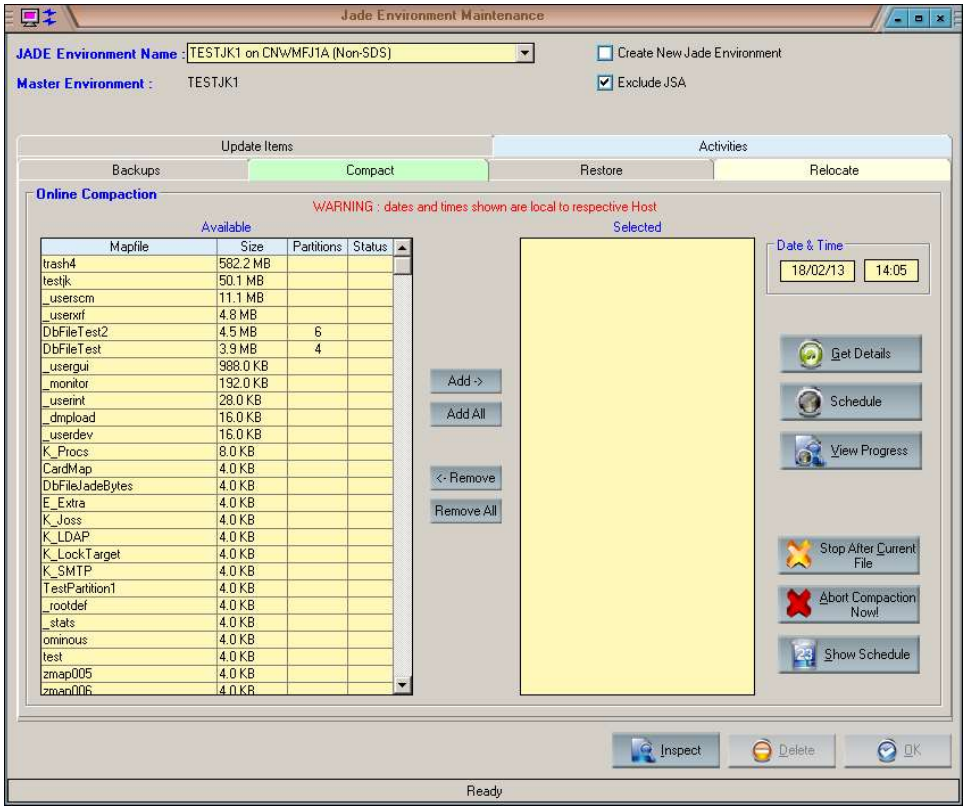
Data retention & Scheduling Information.

Scheduling of the OS Patch Reader application (JossOSPatchMgr) is also controlled via System Global options. This can be defined by selecting the 'Scheduling' tab. The daily OS Patch Reader process is disabled by default and is enabled by entering a date / time for this scheduled item.

By default, Hot Fix information will be retained for 2 years before being deleted out. The deletion process has been incorporated into the daily Janitor run. The retention period may be changed via System Global options under the 'Data Retention' tab.

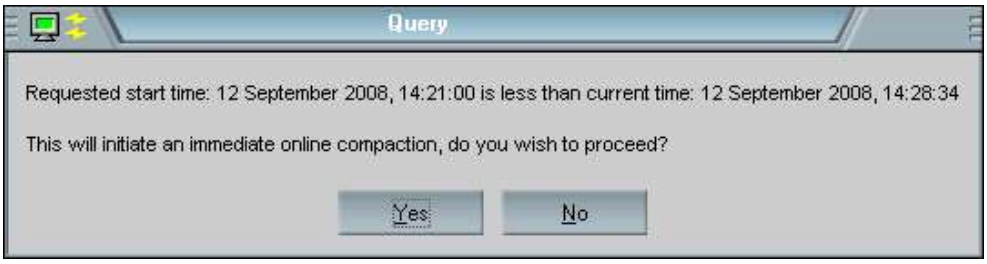
Online Compaction

This optional facility is accessed via the Host Environment Maintenance. From this sheet the user is able to select the mapfile(s) to compact, schedule compactions, cancel scheduled compactions, interrupt a running compaction and view the progress of any active compaction.



Immediate Compactions

Changing to the compaction tab automatically prefills the date and time with the current date and time. Clicking the 'Schedule Compaction' button when the date and time is less than the current date or time is considered an immediate compaction and the user is prompted



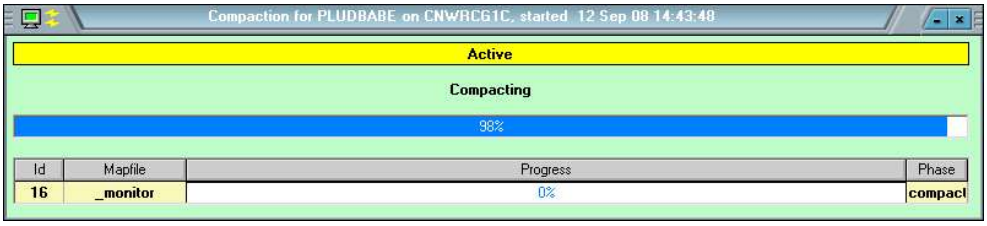
Selecting 'Yes' will result in an immediate compaction. The date and time on the compaction sheet will automatically update to the current date and time. Note that immediate compactions may incur a delay of approx 2 minutes before start.ing.

Scheduling Compactions

If the date and time specified are in the future then this is considered a scheduled compaction. JSM sends a compaction advice to JSA and the compaction is scheduled in JSA. Once the compaction is scheduled, then the compaction runs completely autonomously and does not require JSM to be connected.

View Compaction Progress

If a compaction is running for the selected host environment then it's progress can viewed by clicking the 'View Progress' button.



On completion of a compaction, the completion time, elapsed time and the total size of all the processed mapFiles is shown.

On an interrupted compaction the elapsed time is shown and the total size of all files processed prior to the interruption occurring.

Interrupting A Compaction

If a compaction for the selected host environment is running it can be terminated by clicking either the 'Stop After Current File' or 'Abort Compaction Now!' button. The user is prompted for confirmation before the request is actioned.

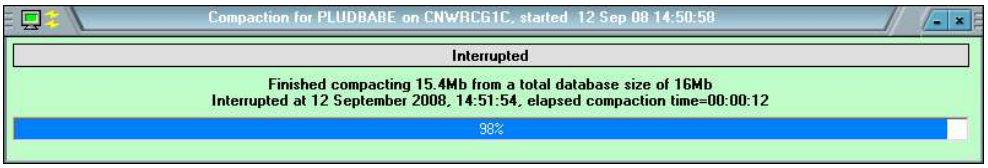
Clicking 'Yes' actions the interruption request.

Stop After Current File

Compaction will continue until the end of the current file being processed. The wait time involved will depend on the file size of the current file and the degree of fragmentation that exists within that file.

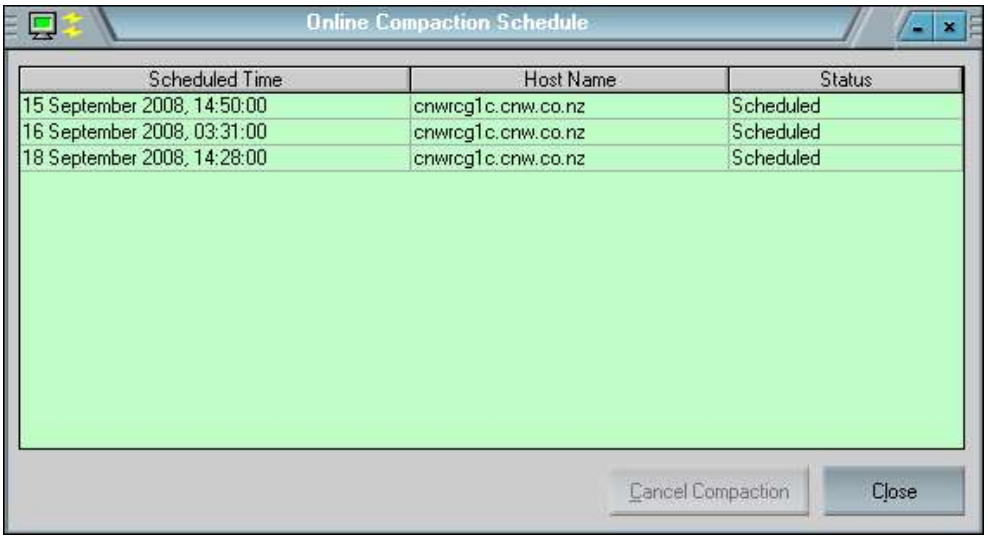
Abort Compaction Now!

Will execute the 'abortReorg' action using JadClient against the environment being compacted and will abort the compaction as soon as possible.

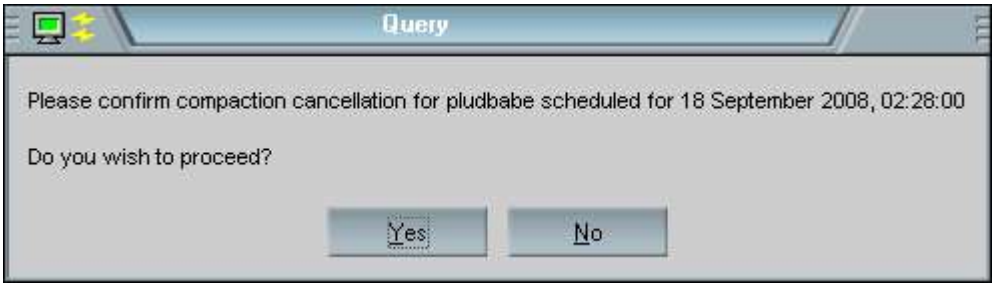


Cancel Scheduled Compactions

Clicking the 'Cancel Scheduled' button presents the 'Online Compaction Schedule' form (which is modal).



Selecting an item in the table enables the 'Cancel Compaction' button. Clicking the button prompts the user for confirmation



Clicking 'Yes' results in a request being sent from JSM to JSA to cancel the scheduled compaction. Should the cancelation be successful JSA sends back a confirmation of the cancellation and this is reflected with 'Cancelled' text in the status column. If the cancellation is unsuccessful then the status column will contain 'Failed' text.

Compaction History

Compaction history can be viewed through the Log History Form

Perquisites for using Online Compaction

- Server Host JSA level must be 3.3 or greater
- Target Environment JADECare Start (CardSchema) must be 3.3 or greater
- Target Environments initialization file setting **ReorgWorkDirectory must** contain <default> or the same path as the database path. This is checked for when the compaction starts and a failure will occur if the pathname falls outside this requirement.

  For environments on JADE 6.3 and later this key is checked in the **[JadeReorg]** section first. If this key is found in the **[JadeReorg]** section then it takes precedence over the same key if present in the **[PersisentDb]** section.

Additional Online Compaction Info

- Multi worker support for online compactions is not available so only one compaction worker is available at this time.
- Progress information by mapfile is not available at this juncture. Progress events returned at the start and end of each mapfile compaction.

JSM Batch Interface

The JSM application is known as **JossBatch.** This application allows a number of batch functions to be executed using the command line interface from sources external to JSM. It produces standard output of what it is doing and will set the exitcode to -1 if an error is detected. The application is executed using the following command line syntax:

```
<JSM binpath>\jadclient path=<JSM sys path> ini=<JSM inifile>
schema=JossControl app=JossBatch server=multiuser endJade <request
and arguments>
```

Application Monitoring Update

Introduced JSM 4.1.01.

Provides the ability to switch one or more applications for one or more JADE Environments on one or more servers from being ‘monitored’ to ‘not monitored’ and vice-versa.

<request and arguments>:

- `action=ApplicationMonitoringUpdate file=d:\tstJSM\server\c_misc\_applMonitoringUpdate_file.txt`

The file variant can contain multiple lines with each line expected to contain the following items delimited by comma:

- <env name>
- <server name – fully qualified domain name>
- Schema Name of application to be or being monitored
- Application Name of application to be or being monitored
- Definition number (1 for non-SDS or Primary, 2 for Secondary)
- Monitored value (true or false)

Note that all values are validated before any updates are performed. All required JSAs performing the monitoring will be updated accordingly.

Example:
```
exmenv,exmserver.exm.co.uk,CardSchema,AppServerMonitor1,1,false
```

The example shown, if validated, would turn off monitoring for application CardSchema,AppServerMonitor1 for environment exmenv on server exmserver.exm.co.uk

Change XCF Parameters

This new function provides a command line methodolgy to update various attributes for a specified XCF item.

To identify the XCF item concerned, the following values must be provided:

- source computer
- source directory
- destination computer
- destination directory

Any one or more of the following attributes can be modified:

- source computer
- source directory
- destination computer
- destination directory
- disabled status

<request and arguments>:

```
action=changeXcfParameters
       sourceComputer=<source computer>
       sourceDirectory=<source directory>
       destinationComputer=<destination computer>
       destinationDirectory=<destination directory>
       [modifySourceComputer=<new source computer>]
       [modifySourceDirectory=<new source directory>]
       [modifyDestinationComputer=<new destination computer>]
       [modifyDestinationDirectory=<new destination directory>]
       [disabled=<true|false>]
```

The arguments sourceComputer, sourceDirectory, destinationComputer and destinationDirectory are mandatory to identify the required XCF item. The values specified are case-insensitive.

At least one of the update arguments must be specified.

NOTE : There is NO validation of the update arguments specified.

Check Backups

Checks backup logs and creates a Tracker call should any backups still be marked as Scheduled or Pending.

<request and arguments>:

```
action=checkBackups
```

Check Connected

Introduced JSM 4.1.01

This function gives the ability to check whether all servers are connected to JSM (via JSA) – exitcode of 0. Where a server name is invalid, or no JSA can be located for a server or any JSA detected is not active, the log will provide details, the standard output will issue an error message and the exitcode will be -1.

There are two variants provided:

- Server=<fqdn>
- File=<fully qualified file name>
  where the file can contain multiple FQDNs, one per line.

<request and arguments>:

- `action=checkConnected server=tstxml1a.tst.co.nz`
- `action=checkConnected file=d:\tstJSM\server\c_misc\_checkConnected_file.txt`

The file variant can contain multiple lines with each line expected to contain a fully qualified server domain name as the server is known to JSM.

Delete JADE Environment

This request will cause the specified JADE Environment to be logically deleted and, eventually, physically deleted.

<request and arguments>:

```
action=deleteJadeEnvironment server=<fully qualified domain name>
environment=<JADE environment name>
[deleteMasterEnvironmentIfLast=<true|false>]
```

Arguments ‘server’ and ‘environment’ are mandatory.

The argument ‘deleteMasterEnvironmentIfLast’ is optional and will default to true. If this argument is true then if JADE Environment specified is the last database entry defined for the Master Environment, then the Master Environment will also be deleted.

Delete Master Environment

This request will cause the specified Master Environment and all associated JADE Environments, to be logically deleted and, eventually, physically deleted.

<request and arguments>:

```
action=deleteMasterEnvironment environment=<master environment
name>
```

Argument ‘environment’ is mandatory.

Delete ODBC Environment

This request will cause the specified ODBC Environment to be logically deleted and, eventually, physically deleted.

```
<request and arguments>:

action=deleteODBCEnvironment host=<ODBC host name> instance=<ODBC
Instance name> environment=<ODBC environment name>

All 3 arguments are mandatory.
```

Delete Server Host

This request will cause the specified Server Host and all associated entities (JADE Environments,. ODBC Environments, .Net Services, Websites), to be logically deleted and, eventually, physically deleted.

```
<request and arguments>:

action=deleteServer server=<fully qualified domain name>
```

Argument 'server' is mandatory.

Extract JARI Details for Environment

This request will extract the registered product releases and their details into a specified file:

```
<request and arguments>:

action= extractEnvJariDetails paramfile=<parameter file>
outputfile=<output file name>
```

where the :

<parameter file> - is a simple file name that is expected to be in <JSM>\server\c_misc directory and where each line is expected to be a customer code and environment name separated by a comma.
Example : cnw,cnwexampl would extra the JARI details for environment cnwexampl for customer cnw. file name includes the path and suffix must be csv or txt.

<output file name> - is a simple file name that will be extracted to <JSM>\server\c_misc directory.

Extract Product Releases

This request will extract the registered product releases and their details into a specified file:

<request and arguments>:

```
action=extractProductReleases file=<file name>
```

where the file name includes the path and suffix must be csv or txt.

Generate JARI Batch Build Files

This request will generate JARI Batch Build files using a source Masterfile (suffixed 'jrim' for each of the environments defined for a specified environment type (as defined in the Master Environment Maintenance form).

Files will be generated for each environment where there is at least one deployable database environment (not disabled and role is non-SDS or Primary).

<request and arguments>:

```
action=GenerateJariBatchBuildFiles type=<type> masterFile=<fully
qualified file name>
```

where type can be:

- JMT
- JTE
- JTE_Identity

The generated JARI Batch Build files will be placed into the same directory as the Masterfile input is located and the environment name will prefix the Masterfile file name and be suffixed with .jri. See log example below for file names

Any exitcode other than zero indicates a failure.

Details are logged into <logs directory>\joss_batchcontrol_<ccyymmdd>.log.
Log example:

```
2020/11/19 08:07:53.290: JossBatch/131: ------------------ Application start
up  -------------------------
2020/11/19 08:07:53.291: JossBatch/131: Application running in TestMode
(Level 2)
2020/11/19 08:07:53.542: JossBatch/131: Now processing requested action
'generatejaribatchbuildfiles'...
2020/11/19 08:07:53.583: JossBatch/131: Creating JARI batch build files for
Jade Third Eye Environments
2020/11/19 08:07:53.583: JossBatch/131: Source is
M:\_Changes\20201118\jaribuild_8_1_10.jrim
2020/11/19 08:07:53.590: JossBatch/131: jtetest JARI batch build file
M:\_Changes\20201118\jtetest_jaribuild_8_1_10.jri
2020/11/19 08:07:53.591: JossBatch/131: tstpjte JARI batch build file
M:\_Changes\20201118\tstpjte_jaribuild_8_1_10.jri
2020/11/19 08:07:53.591: JossBatch/131: Created 2 JARI batch build files for
Jade ThirdEye Environments
2020/11/19 08:07:53.598: JossBatch/131: --------------  Application close
down  ------------------->
```

Get Property Value

For details see section ' Actions Common to JossBatch and KarmaBatchInterface' below.

Global Parameter Distribution

Initiates global parameter distribution to all servers or to the servers for a specified customer

<request and arguments>:

```
action= initiateGlbParamDistribution [customer=<customer prefix>]
```

Jari Status

Provides the ability to get the JARI number and its status.

The 'jari' argument below can define a specific JARI number for the environment or request the status for the latest JARI.

Where the environment and JARI are located, standard output will be issued in form:

'JARI <jari number> for environment '<environment>' status <status> (<status string>)'

The exitcode will be:

- -1 if any error or exception
- <nn> being the status of the sepcified JARI

JARI Status Values:

- 0 Incomplete
- 1 Pending Authorisation
- 2 Pending Rescheduling
- 3 Pending Review
- 4 Pending Restart

- 5 Ready, Scheduled
- 6 Ready, Reviewed, Scheduled
- 10 Deployment Running
- 17 Deployment Aborting
- 18 Deployment Failed
- 19 Completed
- 99 Deleted

<request and arguments>:

```
action=jariStatus environment=<environment id> jari=<jari number>
```

JSM Version Check

Checks that the JSM version defined in the database matches the expected version.

<request and arguments>:

```
action=jsmVersionCheck version=<m.c.rr.ppp>
```

JMT Comparison Build

Added in JSM 4.1.01.

This feature is intended for use where the JADE Master Terminal product is registered. Executing this function will build a comparison file of:

- all JADE< CardSchema and JMT specific files that are expected to be resident and also what the MD5 hash value is registered in JSM
- all JADE< CardSchema and JMT specific files that should have been deleted (based on earlier JMT releases minus the specified release requirements).

The comparison file generated must be specified in the following arguments. The comparison file is expected to be used to compare actual deployed databases and/or remote environments that are maintained using the JMT Deployment function. This file is expected as input to the KarmaBatchInterface function 'JMTCompare', see documentation below.

<request and arguments>:

```
action=JMTComparison_Build environment=<environment name>
version=<target JMT version> file<fully qualified file name>
[jmtOverride=<true|false>]
```

where:

- <environment name> - must match a JSM master environment entity that is defined as a JADE Master Terminal environment type
- <target JMT version>
  - must match a JADE Master Terminal product base release or consolidated release
  - must be equal to or later than the JMT application version currently deployed.
- <fully qualified file name> - the directory must exist and the file name must be Windows compatible.

- jmtOverride argument is optional and defaults to false. If true, then this allows the build to be run for an environment that is NOT marked as a JMT Environment.

Server Patch Status

Extracts servers and outstanding patches count into <JSM log path>\Server_Patch_Status.csv.

<request and arguments>:

```
action=serverPatchStatus
```

Set Up Environment Directories

Provides the ability to create the environment directories on the JSM server (if not already created) and to initialize the version control files

<request and arguments>:

```
action=setUpEnvDirs environment=<environment name>
```

Set Up Jade Environment Directories

Provides the ability to create the target JADE environment directories on the target server (if not already created) and to initialize the version control files to match the values of the JSM environment

<request and arguments>:

```
action=setUpHostDirs environment=<environment name> server=<server FQDN>
```

Shut Down JSM

This request will cause JSM to shutdown.

<request and arguments>:

```
action=shutDownJoss
```

Set Next Deploy Client Distributions

Define the client distributions to be deployed with the next deployment.

<request and arguments>:

```
action=setNextDeployClientDistributions environment=<env id>
clientDistributions=<dist#1>,<dist#2>,<dist#n>
```

Start JARI Batch Build

Introduced JSM 4.0.01. No pre-requisites for JSA.

Provides the ability to specific a batch JARI file (.jri) and to create and auto-initiate a JARI for a specified environment.

Note 1 : One of the following options must be set for this functionality to be usable:

- the global option 'Allow JARI Auto-initiate Via Batch Interface' or
- the specified Master Environment 'Allow JARI Auto-initiate Via Batch Interface' must be set.

Note 2 : The following requirements exist for this feature to be useable:

- The batch script must be initiated from an account that has domain or local admin rights such that it can access the JSM binaries and ini file and execute the JossBatch application
- The jri file and file directories specified in the jri file must :
  - have read/write permissions for the JSM account (under which the application executes) to access them; or
  - the JSM account must have admin rights on the server be locally accessible to the JSM application

The details and any error messages will be logged into:

- JSM's cn_log for application JossBatch
- a log file under a sub-directory \logs in the directory in which the specified .jri file resides. File title is 'jaribuild_<ccyymmddhhmmss>.log

If the JARI build is successful then the .jri file is moved to a sub-directory of \success in the .jri file directory otherwise it is moved to \failure sub-directory.

The exitcode will be:

- the JARI reference number if successful JARI build and initiate.
- -1 if there has been any error or JADE exception.

<request and arguments>:

```
action=startJariBatchBuild jriFile="<filename>"
```

The jriFile specified must include the path name and the file name
i.e. c:\example\example.jri

Switch XCF Group

Introduced in JSM 4.1.01.

Provides the ability to switch a specified XCF group.

<request and arguments>:

```
action=switchXcfGroup group=<XCF group name>
sourceComputer=<current source computer>
```

The <XCF group name> must be a valid XCF group name.

The <current source computer> is a cross-check to ensure the correct group is used. Before a switch is forced the process must find a match of at least one of the 'current' XCF items associated with this group.

NOTE : This action does NOT cause the Global Parameters (and XCF changes) to be issued.

Update JADE Environment

This request will cause the specified JADE Environment to be updated with the values defined.

<request and arguments>:

```
action=updateJadeEnvironment server=<fully qualified domain name>
environment=<JADE environment name>
preInstallBackupPath=<path name for pre-install backup>
standardBackupPath=<path name for all other backups>
[onlineBackupWorkers=<number of workers for online backup>]
[offlineBackupWorkers=<number of workers for offline backup>]
```

```
[tcpPort=<TCP port>]
[deleteOldestBackup=<true/false>]
[databaseLogDays=<number>]
```

Arguments ‘server’, ‘environment’, ‘preInstallBackupPath’ and ‘standardBackupPath’ are mandatory.

The arguments ‘onlineBackupWorkers’ and ‘offlineBackupWorkers’ are optional and will default to 1.

The argument ‘tcpPort’ is optional and has no default value.

The argument ‘deleteOldestBackup’ is also optional and defaults to ‘false’.

The argument ‘databaseLogDays’ is optional and if not provided will retain the existing value

Validate and Extract Interested Parties

This parses the ‘Interested Parties’ list defined for each server and identifies whether the email address is associated with an operator or role entity within JSM. The email address, the server(s) it is associated with and if it does not match a JSM entity are extracted to a csv file <misc path>\InterestedParties_Extract.csv

<request and arguments>:

```
action=validateExtractInterestedParties
```

JSA Batch Interface

The JSA application is known as **KarmaBatchInterface.** This interface will allow a number of batch functions to be executed using the command line interface from sources external to JSA. It produces standard output of what it is doing and will set the exitcode to -1 if it an error is detected.

If JSA has been initialized and has not yet connected to JSM, then any execution of **KarmaBatchInterface** will fail with message:

```
ERROR: JSA has not yet completed initialization.
```

The application is executed using the following command line syntax:

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
<request and arguments>
```

```
Example of using the interface to invoke an ad-hoc backup:
```

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
action=runadhocbackup name=<Jade environment name> backupworkers=3
backupdir=<directory name>
```

```
To run the backup offline, specify argument offline=true
```

```
To specify an ODBC environment to be backed up, specify the ODBC
instance in addition to the database name:
instancename=<instance name>
```

```
If it is a remote ODBC environment please provide host name too:
host=<remote host name>
```

JadeAdmin and Configuration

Create Configuration

With the implementation of regime 3, the standard run of JadeAdmin using **createConfiguration** does not create a true regime 3 version of the configuration

file. Also the content of the dirmaps sections is dependent on which version of cnwplugin being used. Version 2.0.0.41 and later create a pseudo regime 3 formatted file and earlier version create a true regime 2 format.

The batch interface now provides a mechanism to create an appropriately formatted configuration file. If the <request and arguments> does not contain a 'regime=<x>' specification, then the regime is derived from the host environment object in JSA (which is derived from JSM's repository)..

<request and arguments> are as follows:

```
action=createConfiguration <host env name> [regime=<regime number>]
```

where the regime argument is optional

Update Configuration

The batch interface now provides a mechanism to update an existing configuration file. Running this command will clean up any unrequired entries in the config file and add required entries that aren't present. The regime is automatically derived from the host environment object in JSA (which is derived from JSM's repository).

<request and arguments> are as follows:

```
action=updateConfiguration <host env name>
```

You would not normally be required to run this command. It is intended for operations staff to correct any configuration files where some sort of deployment failure has occurred and manual intervention is required.

Setup and Create Configuration

This allows the creation of a configuration file for a JADE Environment that has not yet registered in JSA. The parameters required are minimum necessary to create temporary object instances within JSA in order to create the configuration file. The arguments are all required.

<request and arguments> are as follows:

```
action=setupCreateConfiguration env=<master environment name>
    jadeEnv=<JADE environment name> databaseDrive=<volume letter>
           databaseRole=<0 | 1 | 2> regime = <2|3> name=<named
section>
```

Cycle JSA Server Application

Provides the ability to stop (if running) and start a JSA server application. As at the 4.0.02 release valid JSA server applications that can be stopped and restarted are:

- KarmaAlertSetHandler
- KarmaCommsMgr
- KarmaGeneralEventMgr
- KarmaStatsCollector

<request and arguments> are as follows:

```
action=cycleJSAServerApplication <application name>
```

Dump Task Schedule

Extracts all the Task Scheduling details defined in JSA. Used for debugging and analysis. The file produced is

dump_Task_Scheduling_Info_<ccyymmdd>_<hhmmss>.txt and is written to the logs directory.

<request and arguments> are:

```
action=Dump_TaskSchedule_Info
```

Get Property Value

For details see section ' Actions Common to JossBatch and KarmaBatchInterface' below.

Hot Fix Update Product Repository

There are times when a JSM hot fix is required to update the JADE and/or CardSchema product repository with a patched file.

This requires one or more files to be replaced and for the registered item to be updated with new file version, modified timestamp and MD5 hash value.

The product repository is only updated if:

- An exact match is found with the hotfix file and its corresponding entry in the product repository AND
- If the MD5 values differ.

<request and arguments> are:

```
action=hotFixUpdateProductRepository
```

An exitcode of -2 means that there were no updates detected.

Initialize ODBC

Using the initialize ODBC function causes all ODBC Entities to be deleted and for the ServerHost.glbParamVersion to be set to null. This will cause:

- All ODBC entities to be re-created when JSA connects with JSM
- All ODBC and JADE Environments to have their backup specifications updated.

NOTE : This option must only be run with single-user being specified.

<request and arguments> are:

```
action=Initialize_ODBC
```

Initiate Discovery

Provides facility for an external agent (outside of JSM / JSA) to initiate the JSA discovery processes. Current intent is for Commander to execute this when cluster resource management fail over is detected.

```
action=initiateDiscovery
```

JADE Environment Is Active

Advises whether the specified JADE environment is active as seen from JSA.

Standard output respone will be 'isActive=true | false'

<request and arguments> are:

```
action=isActive <environment name>
```

JSA Version

This request will return the version identifier of JSA as standard output in the form **JSA Version = 3.5.01.011**.

<request and arguments> are:

```
action=JSAVersion
```

JMT Compare

Added in 4.1.01.

This function takes the file produced by JossBatch function 'JMTComparison_Build' and does a comparison of required files (including checking MD5 hash values) and files that should no longer be present as defined by the JADE Master Terminal version specified when executing JossBatch.

Where there are inconsistencies detected the 'response' argument identifies whether this is to be logged, create/update a Tracker call or both. Note that where there are inconsistencies the exitcode will still be zero.

When logging is specified, any inconsistencies are logged into JSA's karma_batchinterface_<date>.log. Any runtime errors will be logged into the same log and will cause an exitcode of -1 as well as the error message to be issued to standard ouput.

<request and arguments> are:

```
action=JMTCompare environment=<environment name> file=<fully
qualified file name> [response=LogOnly|AdvisoryOnly|Both]
[debug=true|false] [jmtOverride=<true|false>]
```

where:

- <environment name> must be a JADE Environment name resident on the JSA server
- <fully qualified file name> a file name that exists
- Response argument is optional and if not specified will default to Both such that any discrepancies are logged AND will cause a Tracker to be created / updated.
- Debug argument is optional and will default to false.
- JMT Override argument is optional and defaults to false.
  If this override is set to true then the JADE Environment's master environment does NOT need to be marked as a JMT Environment.

Regardless of whether logging is specified, any files remaining that have NOT been checked as 'required' will be logged. This may indicate files that should not be present

Licence Info

This request will return the licence name and licence key as standard output prefixed by keywords **LicenceName** and **LicenceKey** respectively.

<request and arguments> are as follows:

- `Query=licenceInfo <environment name>`
- `Action=licenceInfo < environment name>`

Run Ad-hoc Backup

A batch interface has been provided that enables a scripted ad-hoc backup to be run of a JSA managed JADE or ODBC environment.

The interface enables specification of the following optional (*) backup parameters:

- backupMode – optional; default behaviour is to perform an online backup.
- backupWorkers – optional; number of backup workers that perform the backup. Defaults to number defined in JSM for this managed environment. If specified this will update JSA values depending on offline setting
- backupDir – optional; final backup directory. Default is backup directory name specified in JSM.
- instanceName – optional; defaults to JADE Environment backup. If specified then backup is for ODBC Managed Environment
- host - optional; if 'instancename' argument provided then the host name defaults to the server that JSA is executing on.
- name – required. This is the name of the database (JADE environment or ODBC environment) that is to be backed up.
- compressFiles – optional; defaults to false. Specify compressFiles=true to use compression for backup. This option is meaningless if an ODBC environment backup
- executeScripts – optional; defaults to true. This is only meaningful for JADE Environments and will cause the execution of appropriate scripts (dependent on whether online or offline backup) if they are resident.

NB: The optional 'offline' argument will still be permitted to retain existing functionality for existing scripts. The value may be true or false.This argument may not be specified if the 'backupMode' argument is defined.

Progress of the backup is provided within JSM in the normal manner.

* The values in JSM are used if these optional parameters are not specified. If parameters values are defined then these temporarily override the values defined for the environment in JSM (the values in JSM do not change).

Examples:

The following will execute an ONLINE adhoc backup for JADE Environment on this server, directing the backup to directory m:\backup_area and using 3 workers while backing up the database

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
action=runadhocbackup name=ExampleDb backupworkers=3
backupdir=m:\backup_area
```

The following will execute an OFFLINE adhoc backup for the specified JADE Environment on this server, directing the backup to directory m:\backup_offline, using 4 workers while backing up the database and will suppress the execution of scripts even if they are resident.

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
action=runadhocbackup name=ExampleDb backupworkers=3
backupdir=m:\backup_area backupMode=offline executeScripts=false
```

The following will execute a JOURNALED adhoc backup for the specified JADE Environment on this server, directing the backup to directory m:\backup_offline, using 4 workers while backing up the database and will suppress the execution of scripts even if they are resident.

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
action=runadhocbackup name=ExampleDb backupworkers=3
backupdir=m:\backup_area backupMode=journaled executeScripts=false
```

The following will execute an adhoc backup for remote ODBC Environment accessible from JSA on this server, directing the backup to directory 'arntesting'.

The arguments 'backupWorkers', 'backupMode' and 'offline' are superfluous in the context of this backup and will be ignored if provided.

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
action=runadhocbackup host=cnwNativeTest instanceName=. name=Test1
backupdir=arntest
```

Run Ad-hoc Compaction

A batch interface allows for a scripted ad-hoc compaction to be run for a JSA managed JADE environment.

The interface enables specification of the following compaction parameters:

1. name – required. This is the name of the JADE Environment database that is to be compacted
2. compactWorkers – optional. At this juncture this will always be overridden and set to 1. A future change to this implementation will see this limitation removed.
3. compactFiles – required. At least one map file must be specified. A list can be provided separated by commas.

Examples:

The following syntax will execute a compaction of mapfiles UserMap and Transactions for database ExampleDb.

```
<JSA binpath>\jadclient path=<JSA sys path> ini=<JSA inifile>
schema=JossControl app=KarmaBatchInterface server=multiuser endJade
action=runAdhocCompaction name=ExampleDb compactWorkers=3
compactFiles=UserMap,Transactions
```

Possible Error Messages:

```
ERROR: <app name> in exception state, terminating...
```

Unhandled exception occurred, open contact for JadeCare Systems Manager and attach stackdump and cn_logs for JSA.

```
ERROR: Unexpected argument '<argument name>'
```

An argument was provided on command line that is inconsistent with action requested.

```
ERROR: Expected minimum of one mapfile to be specified for
compaction
```

No map file names were provided. Add argument 'compactFiles' and either one or a list of map files to be compacted.

```
ERROR: Expected name of managed environment to be backed up
(name=<db name>)
```

The environment name has not been specified. Add an argument 'name=<database name>' to the command line.

```
ERROR: Specified managed environment does not exist on server
(<fqdn>).
```

The environment name specified is not registered as a database in JSA. Check the name, confirm that it is a database environment and that the environment is defined in JSM for this server.

```
ERROR: Specified environment is not a JADE Environment.
```

The environment name specified is not registered as JADE environment in JSA. Check the name, confirm that it is a database environment and that the environment is defined in JSM for this server.

```
ERROR: Environment database volume is invalid.
```

The environment name specified defines a database volume that is not valid for this server. Check the database volume defined in JSM and your ini files.

```
ERROR: Environment log volume is invalid.
```

The environment name specified defines a log volume that is not valid for this server. Check the log volume defined in your ini files.

`ERROR: <env name> on <server>` has not registered any mapfiles yet!

The environment specified has yet to register any mapfiles. Ensure that database environment is running and soon after registering the environment as active, the mapfiles will be registered.

```
ERROR: Mapfiles not found: <list of missing mapfiles>
```

One or more of the map files that you have specified do not match the actual map files registered by the application. Review the map files defined after argument 'compactFiles'

Request Update Thinclient Files

This provides a facility for any external process that restores/refreshes the thinclient download files (ie a full restore, the REFRESH process) to request, via JSA, that JSM update the thinclient download files.

This is to avoid conflict on the next deployment when the environment is regime 3 where thinclient downloads are checked.

The request will only be forwarded to JSM if the local JSA can locate the specified JADE Environment and its master environment has specified 'regime 3'.

```
<request and arguments>
```

```
action=update_ThinClient_Files <JADE environment name>
```

The application will provide standard output to identify what is happening and return an exitcode of 0 if the request is acceptable. Failure will be identified by an exitcode other than 0.

Exitcode of 0 will be returned if:

- The deployment regime is 2 (and request will NOT be sent to JSM)
- The deployment regimes is 3 and the message was sent to JSM

Exitcode greater than zero will be returned if:

- Local JSA is not able to locate the JADE Environment specified
- JSM is not connected
- Any other unhandled exception is incurred.

Validate Ini File

This function will validate the contents of a target environment's ini file. The first implementation only checks the following items and that they are syntactically correct:

- [JSMDeployment] ExcludeFile entries
- [JSMDeployment] RemoveeFile entries
- [JSMRestore] ExcludeFile entries

```
<request and arguments>
action=validateIniFile <JADE environment name>
```

The application will issue standard output of any errors found and return an exitcode of 0 if the ini file content checked is valid. Failure will be identified by an exitcode other than 0.

Exitcode greater than zero will be returned if:

- Local JSA is not able to locate the JADE Environment specified
- The ini file is not syntactically correct
- Any other unhandled exception is incurred.

Run Environment Restore

A batch interface allows for a scripted ad-hoc restore of a JADE environment from a backup.

The interface enables specification of the following restore parameters:

1. name – required. This is the name of the JADE Environment database that is to be restored
2. backupdir – required. The path to the backup root directory
3. lastjournal – required. The last journal to be recovered
4. restoreworkers – optional. Defaults to backup workers specified for the environment (or 1)
5. restorebin – optional. Specifies whether binaries are restored. Value true or false, default is true
6. restoremisc – optional. Specifies whether the misc directory is restored. Value true or false, default is true
7. clearfilepaths – optional, default false
8. clearsdsrole – optional, default false
9. differentenvironment – optional, default false
10. useexistingjournals – optional, default false
11. restorenorecovery– optional, default false

Examples:

The following syntax will execute a restore of environment test1.

```
D:\cnwkarma\server\c_bin\jadclient path=D:\cnwkarma\server\c_system
ini=D:\cnwkarma\server\c_bin\cnwkarma.ini schema=JossControl
app=KarmaBatchInterface server=multiUser endJade action=runRestore
name=test1 backupdir=D:\cnwbackup\test1_23Mar restorebin=false
lastjournal=289
```

Actions Common to JossBatch and KarmaBatchInterface

Get Property Value

Provides the ability for a JSM / JSA batch process to obtain the value from any property for instances of the following classes:

- Root
- ServerHost
- Environment
- JadeEnvironment
- ODBCHost
- ODBCInstance
- ODBCEnvironment

The value is returned as the process exit code when:

- The property is type Boolean and value is converted to Integer
- The property is type Integer
- The property is type Integer64 and the value is not larger than Max_Integer.

If the property type does not match any of the above the exitcode returned is -2 (output to file) and the result is output into a file in the logs directory for JSM / JSA with title
'getPropertyValue_<propertyName>_<className>_<entity name>.txt"
which is formatted as follows:

- Line 1 – entity name
- Line 2 – class name
- Line 3 – property name
- Line 4 – property type
- Line 5 – property value (converted to String)

NOTE : The property name is case-sensitive.

Syntax:

```
<binpath>\jadclient.exe ini=<ini file> path=<system path>
schema=JossControl
app=[JossBatch | KarmaBatchInterface] endJade
action=getPropertyValue
class=<valid class name -see above>
propertyName=<JADE property name (case-sensitive)>
[environment=<environment name>]
[server=<server FQDN>]
[host=<ODBC host name>]
[instance=<ODBC Instance name>]
```

Example #1:

```
action=getPropertyValue
class=JadeEnvironment
propertyName=deleteOldestBackup
environment=TestJK10
server=cmfj1f.cnw.co.nz
```

In this case, the property is a Boolean and is true so the process exit code is 1.

Example #2:

```
action=getPropertyValue
class=JadeEnvironment
propertyName= preDeployBackupRetentionDays
environment=TestJK10
server=cmfj1f.cnw.co.nz
```

In this case, the property is an Integer with a value of 5 so the process exit code is 5.

Example #3:

```
action=getPropertyValue
class=JadeEnvironment
propertyName= emailAddress
environment=TestJK10
server=cmfj1f.cnw.co.nz
```

In this case, the property is a String with a value of 'exampleEmail@gmail.com' so the process exit code is -2 and file getPropertyValue_emailAddress_JadeEnvironment_TESTJK10 on CNWMFJ1F (Non-SDS).txt. was created in the logs directory and contained the following:

```
Class        : JadeEnvironment
Entity       : TESTJK10 on CNWMFJ1F (Non-SDS)
PropertyName : emailAddress
PropertyType : String
exampleEmail@gmail.com
```

KarmaTimeChecker

Application Type: Non-GUI

Optional application, if required this can be launched manually using the **KCStartAppInNode** application or it can be defined as a Server Application in the ini file.

This application starts a 5 second timer and can be used to check if there is any large variances in the server's time (which is often symptomatic of more than one time server being used to synch the server). Where a variance of 15 second or more occurs, an alert is generated to JSM.

ODBC Environment

ODBC Environment Directory Structure

The ODBC Environments to be **managed** by JSM/JSA must conform to a simple set of standards with regard to directory names.

The ODBC Environment directory naming is also dependent on the instance name. If the instance name is the default (eg . – period) then the parent directory node will be derived from the database type. For MSSQL instances this will be MSSQL. Otherwise the instance name is used for the parent directory node.

Current ODBC Environment Directory

Each ODBC environment resident on a managed server is defined as having an environment drive and a logs drive which can be the same or different.

If the environment drive is d, the log drive is e and the environment name is SQL_TEST then:

- If the default instance was used then the environment directory would be d:\MSSQL\SQL_TEST and the log directory would be e:\MSSQL\SQL_TEST
- If the instance name is 'example_instance' then the environment directory would be d:\example_instance\SQL_TEST and the log directory would be e:\example_instance\SQL_TEST

Within the ODBC environment directory structure we expect to find the following sub directory structure:

<ODBC environment root directory>
-\data
-\misc
- \scripts

Should pre-backup and/or post-backup scripts be required, these can be placed in <ODBC environment root directory>\misc\scripts. These will be executed if resident.

Backup ODBC Environment Directory

The ODBC environment backup directory path should be defined in similar way as is done for JADE environment eg t:\cwnbackup.

For a managed environment on a managed server, in the same way that the directory structure is derived for the current ODBC environment, so too is the backup path.

If the backup path root is defined as t:\cnwbackup then the following backup directories will be utilized:

- If the default instance was used then the back directory would be t:\cnwbackup\MSSQL\SQL_TEST
- If the instance name is 'example_instance' then the environment directory would be t:\cnwbackup\example_instance\SQL_TEST

For ODBC Environments resident on a non-managed server (ie an Amazon RDS database) then the standard backup directory is totally free format.

'System' ODBC Instance Directory

Each ODBC Instance has one or more 'system' definitions (for MSSQL these are MASTER, MODEL and MSDB). Within JSM/JSA these are identified with a generic ODBC environment '_system'. This name is also used in the derivation of the backup directory path used and the backup image is a concatenation of the 3 backups (for MSSQL).

If the backup path root is defined as t:\cnwbackup then the following backup directories will be utilized:

- If the default instance was used then the back directory would be t:\cnwbackup\MSSQL_system
- If the instance name is 'example_instance' then the environment directory would be t:\cnwbackup\example_instance_system

Monitoring

JSM supports basic monitoring functionality for non-JADE databases by utilising the ODBC standard. There are some pre-requisites to using this functionality

- JSA version must be v3.5.02 or later
- cnwsql.dll (v1.0.0.6 or later) must be present in the JSA binaries directory

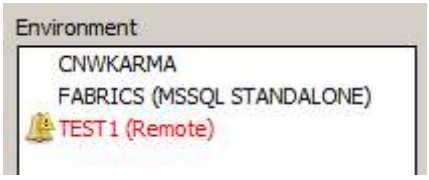
WARNING : *The first implementation of this interface only supports SQL databases. A future release will expand the ODBC monitoring capabilities to other databases which provide the ODBC interface.*

When ODBC monitoring has been enabled JSA will attempt to open a ODBC connection to the monitored environment based on a user specified time interval.

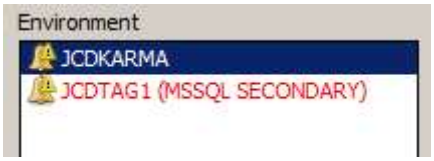
If no connection can be established an alert (or a JADECare Tracker if instantiated) will be raised. This alert will repeat on the expiry of every time interval.

For example, if you specify a 60 second interval then every 60 seconds the ODBC monitored environment status is checked. A repeat alert will be raised every 60 seconds until such time as the ODBC environment becomes available again or monitoring is switched off. If the ODBC environment becomes available the alert will automatically clear.

System View will also reflect the running state of ODBC monitored environments. The various states look like this:



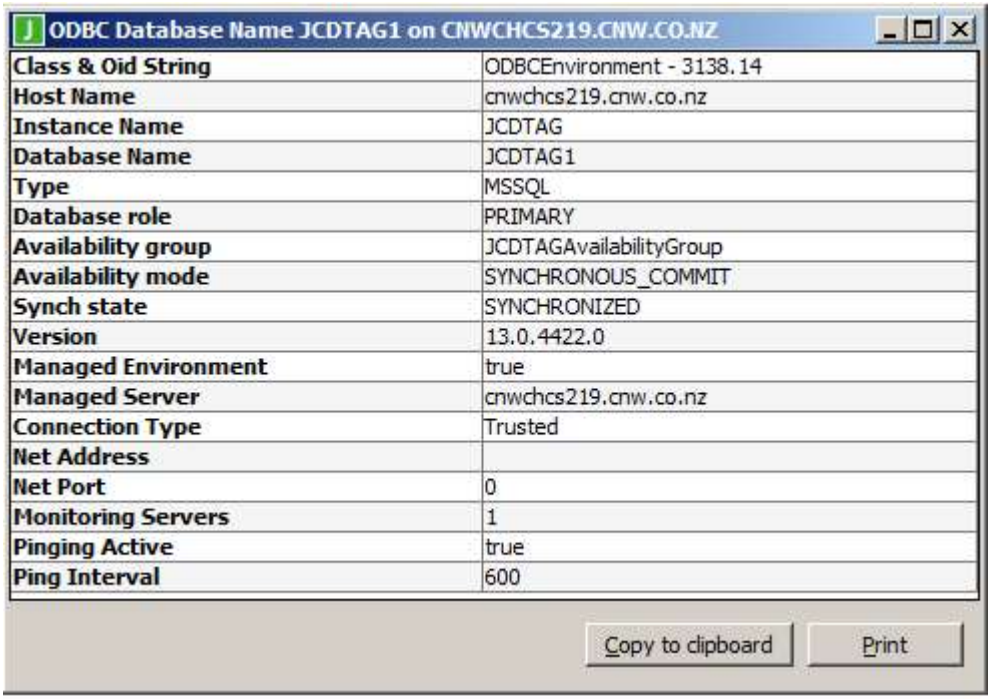
Fabrics is responding



JCDTAG1 is alarming

Fabrics is not being monitored

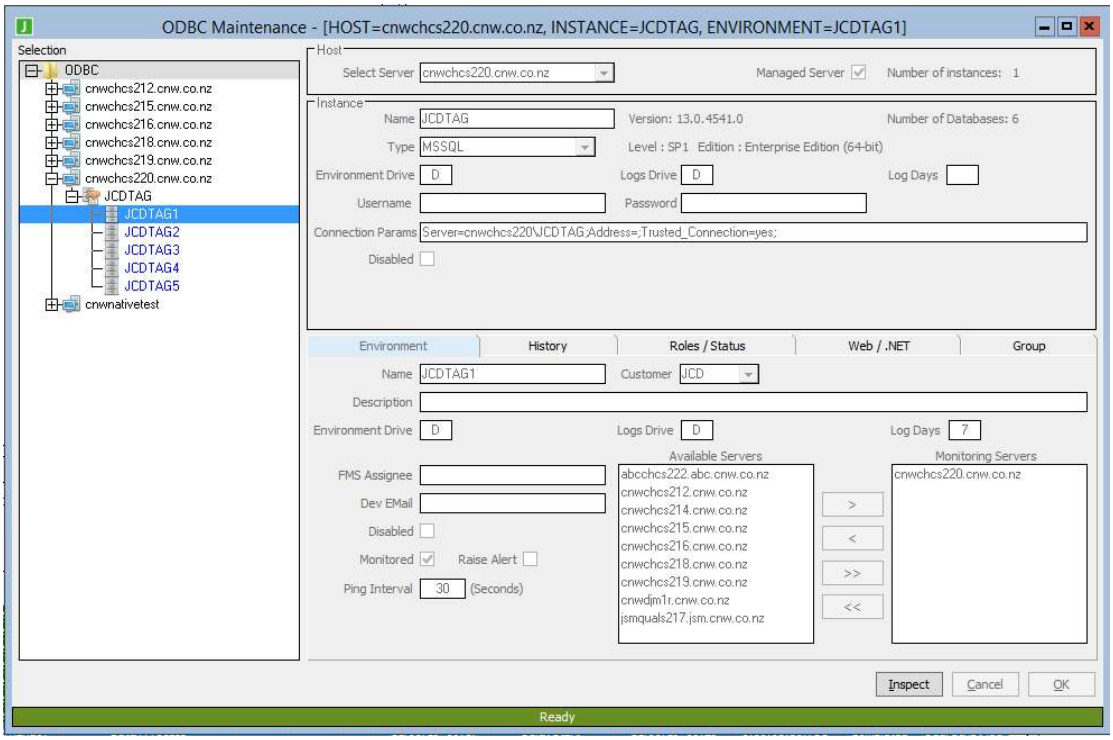
Double-clicking an ODBC environment shows the basic configuration elements in the System View Summary form:



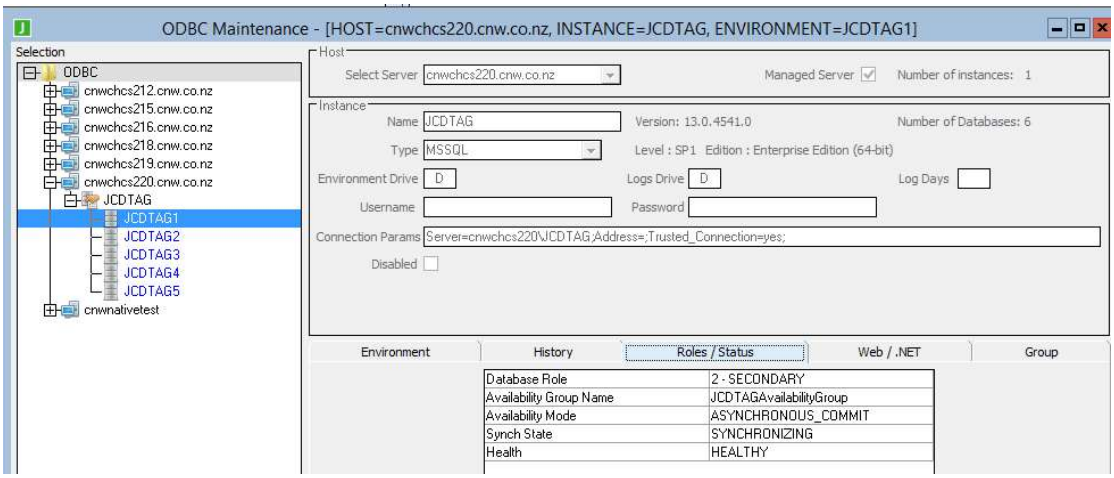
Right mouse clicking on an ODBC environment brings up some menu items:

Clicking the 'ODBC Environment Details' item will display the System View Summary form as shown above.

Clicking the 'ODBC Environment Setup' item will display the configuration maintenance form for this ODBC environment:

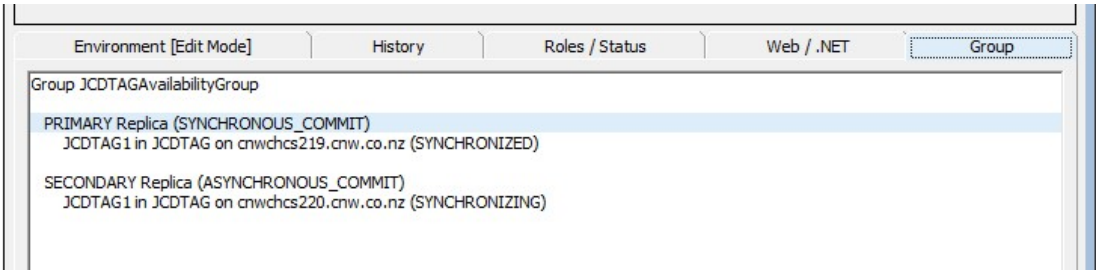


The Roles/Status sheet displays database role and status information:



The History sheet displays the history of database role changes (for example when a role switch from Mirror to Principal occurs).

The Group sheet displays availability group information for a replica database:



ODBC Monitoring Configuration

The ODBC Maintenance form can be accessed in two ways, either from the popup menu item for the environment in System View or via the 'Setup / ODBC Maintenance' menu item.

In order to monitor ODBC environments you must define the following items:

- Host (where one or more ODBC instances reside)
- Instance
- Database

These three values are used by JSM to form the connection string used to connect to the monitored database.

Defining an ODBC Host

Start by defining a Host. Click 'ODBC' then use the popup menu item 'Add a new ODBC host'

The 'Host' area will become editable:

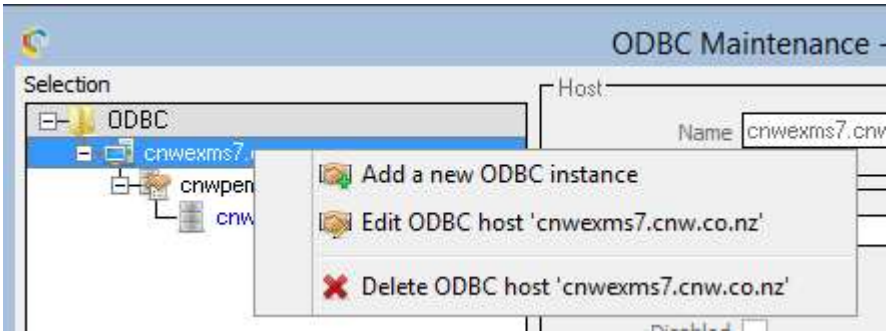
The ODBC Host can be defined as either a JADECare Managed Server (ie it has a JSA executing on it and the server and JSA have registered in JSM) or a non-JADECare Managed Server which has one or more ODBC Environments that are monitored remotely.

If the server is a JADECare Managed Server, select a managed server from the drop-down combo-box and click the 'OK' button (the 'Managed Server' check box is checked by default and this is appropriate).

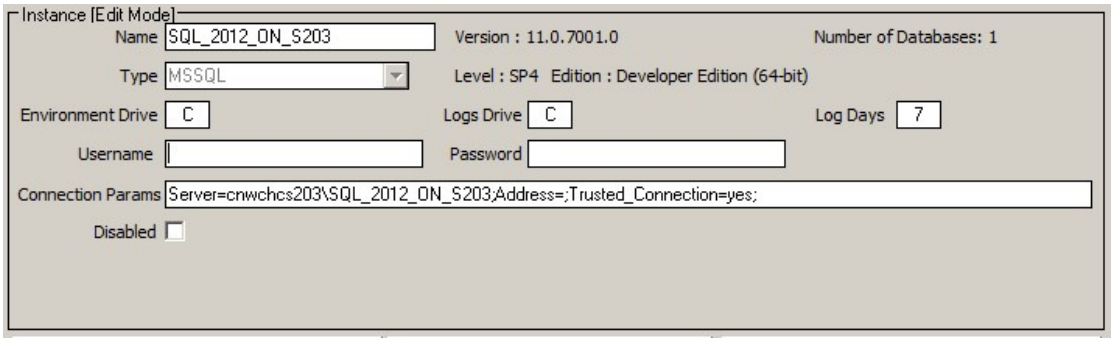
If the server is a non-managed server (for example an Amazon server hosting an RDS database), uncheck the 'Managed Server' checkbox and enter the ODBC host name and click the 'OK' button. For RDS databases, the host name must be unique but is not used to define the connection to the database. For RDS databases (for example those hosted by Amazon), the net address and port specifications on the ODBC Instance pane are used to connect to the database.

Defining an ODBC Instance

Now the host has been defined you can add an 'Instance'. Use the popup menu items on the newly defined host:



Click 'Add a new ODBC instance' and the 'Instance' area will become editable:



Note: If this is the default instance for your ODBC databases you must enter '**.**' (period) as the instance name. If however you have multiple instances you would name them in this field accordingly.

- Environment Drive

  This should identify the volume on which the ODBC environment is located

- Database Type

  Selecting the database type for an instance determines the backup script applicable to this instance (the backup script has the same name as the type to which it applies). The type is used for display purposes in forms such as System View, System View Summary, ODBC Environment Summary and other forms.

- Logs Drive

  This should identify the volume on which ODBC Environment logs are to be located.

- Log Days

  This identifies the retention period of ODBC environment logs

- Connection Params

  Specify connection parameters passed in the connection string when connecting any ODBC database in the instance. This connection string is used by monitoring JSAs that predate 4.0.03 and also to connect to system databases that are backed up (for example tempdb in MSSQL). From 4.0.03, monitoring JSAs use the connection string defined in the environment section (see details below).
  The environment monitoring JSA will contruct a connection string specifying the name of the database to connect to and what ODBC driver to use (if none is specified) and append the parameters specified in this field. Example Connection Param strings are:

This string corresponds to the database in the picture above:
Server=cnwchcs203\SQL_2012_ON_S203;Trusted_Connection=yes

or, for an RDS database on a non-managed server:
Address=<some address at amazonaws.com>,<port>; Trusted_Connection=no

- Username

  A valid username with access rights to the database. The user can be defined locally in the database or be a valid windows account (local or domain) with database permissions if the database is configured to do windows authentication.

- Password

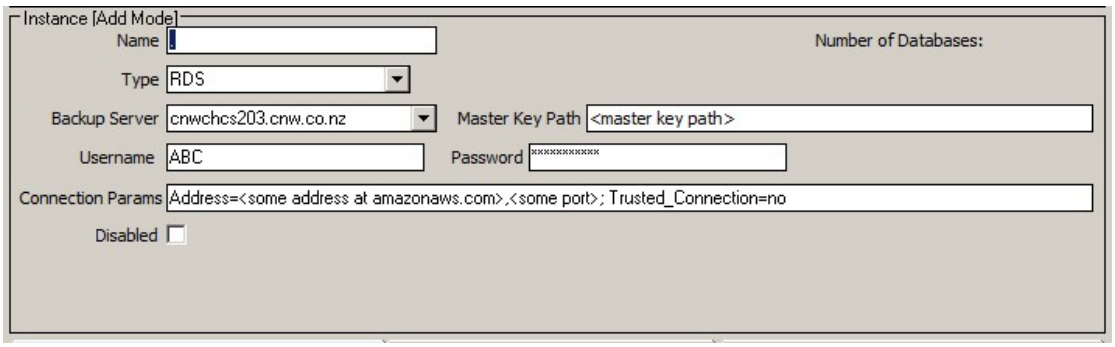
  A valid password for the 'Username' being used.

- Disabled

  This allows the instance and all of its ODBC environments to be disabled. Note that if the instance is disabled then all of the ODBC environments within it are also disabled. Marking the ODBC Instance as enabled does not mean that all associated ODBC environments are enabled.

Click the 'OK' button to commit the new instance.

A non managed instance has additional fields, Backup Server and Master Key Path



- Backup Server

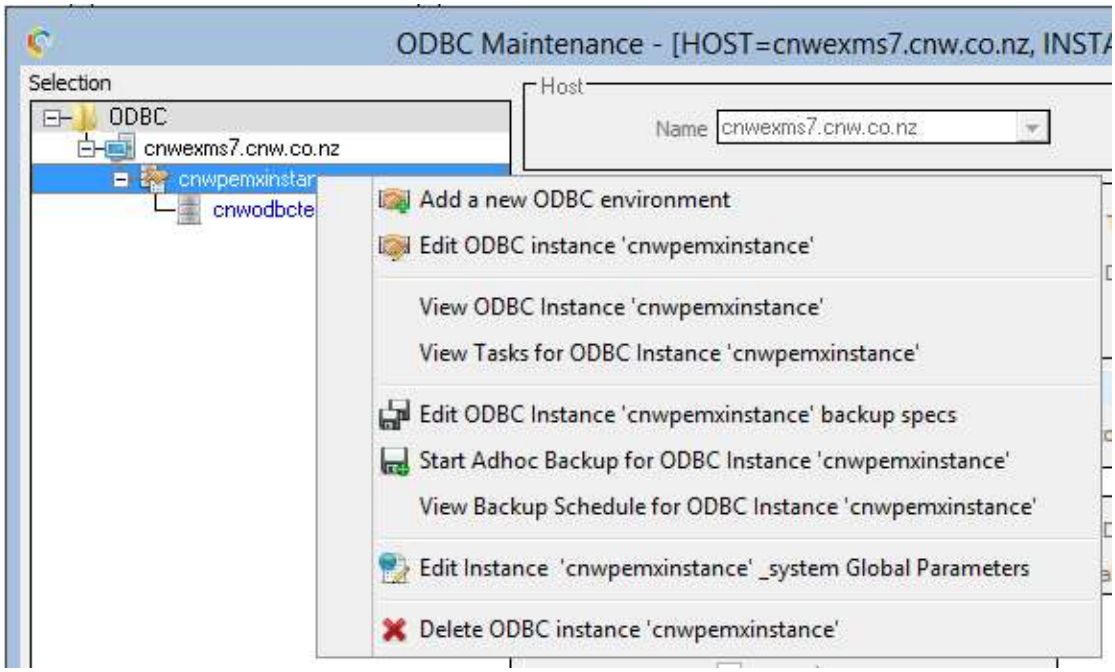
  The nominated backup server is a managed server that is able to back up the instance by executing the backup script applicable to the type of database.

- Master Key Path

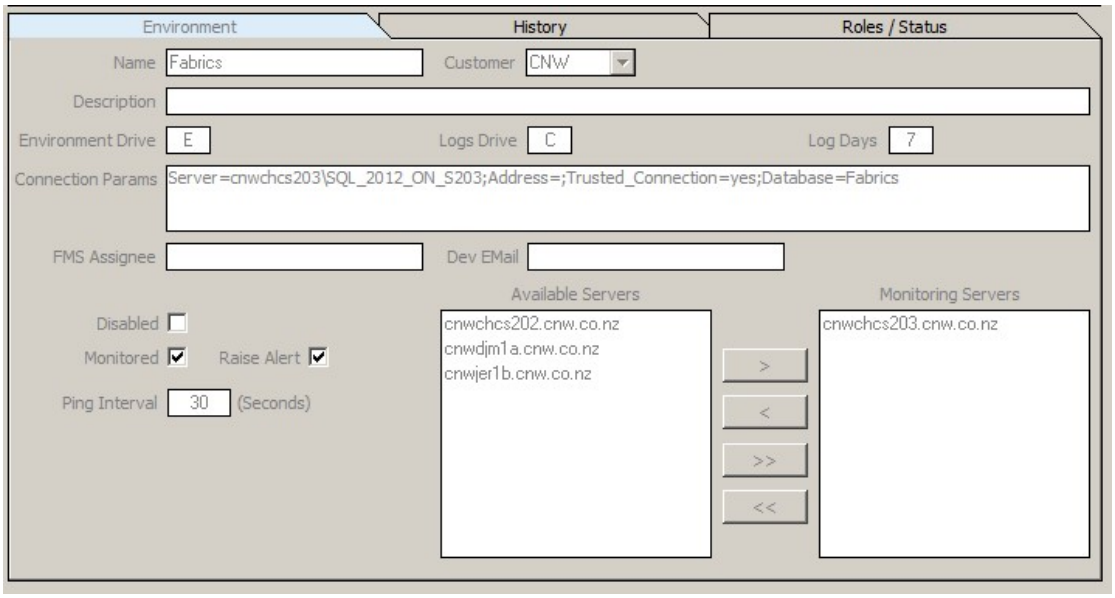
  This field is required to be able to successfully access an Amazon RDS database. It will be provided by the RDS Database Administrator and is passed to the appropriate backup script.

Defining an ODBC Environment

Once an ODBC instance has been defined you can start creating the ODBC environment configuration. Use the popup menu items on the newly defined instance:



Click 'Add a new ODBC environment' and the 'Database' area will become editable:



- Name (Required)

  The exact name of the ODBC database you wish to monitor.

- Customer (Required)

  The customer id associated with this configuration. This is required by the toolset.

- Description

  This can be anything, enter something descriptive or important concerning this configuration.

- Environment Drive

  This should identify the volume on which the ODBC environment is located

- Logs Drive

  This should identify the volume on which ODBC Environment logs are to be located.

- Log Days

  This identifies the retention period of ODBC environment logs

- Connection Params

  The connection parameters that form the basis of the connection string used to connect to the database.

- FMS Assignee

  If a fault management system is instantiated (JADECare Tracker) then assign faults to this account. Even if no fault system is installed this is useful also for designating a point of contact for this configuration.

- Disabled

  If the server or the instance was disabled then the ODBC environment will be marked as disabled. To enable the ODBC environment requires that the parent items be marked as enabled.
  Marking the ODBC environment as disabled means that backups and task schedules for this environment will be deactivated.

- Available Servers

  These server hosts have a JSA that meet the criteria for monitoring this ODBC database. It is up to the individual defining the database configuration to make sure the server host has network visibility to the ODBC database.

  Multiple hosts can monitor the same ODBC database to offer some redundancy should a server host fail. Note that an alert will be generated for only the Server Hosts that fail to successfully connect to the ODBC database.

  Note: Unlike JADE applications ODBC databases initiate no connection to JSA therefore we are unable to "follow" the ODBC databases around. Monitoring relies solely on successful connections at timed intervals. This is a limitation we have no control over.

- Monitoring Servers

  The JSA on these server hosts will be responsible for monitoring the ODBC database.

- Monitored

  Turn active monitoring on or off.

- Ping Interval

  Determines how frequently the ODBC database status will be checked. Note that the presence of sentinel file pausepinging.htm in the environment's misc directory will cause ping actions to be skipped.

- Ping Wait Time

  Determines how long to wait before getting a response back from the ODBC database. If this 'Ping Wait Time' passes then an alert will be raised.

- Raise Alert

  This is only relevant where JADECare Tracker has been instantiated. When enabled it means not only will a JADECare Tracker Incident be opened/updated but that an alert will be raised also.

  If JADECare Tracker hasn't been instantiated this control is not visible. The default operation for an ODBC alarm advice is:

  Tracker Installed:

  - default action is Tracker Incident created / updated
  - if 'Raise Alert' is enabled then an alert will be created / updated also

  No Tracker Installed:

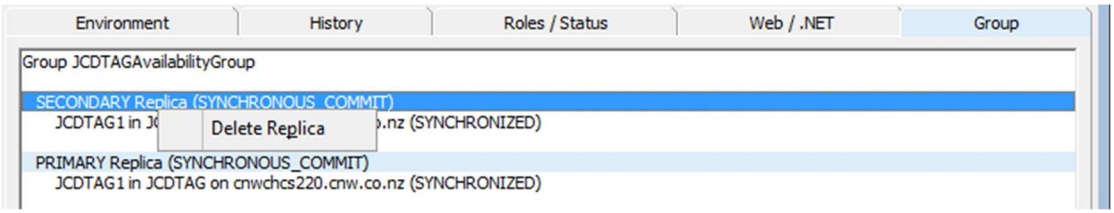
  - default action is alert created / updated

Click the 'OK' button to commit the new database.

Delete a Database Replica

This option is provided so that the replica definition can be removed from JSM. This operation is only necessary if the databases in the replica are not pinged again (if all of the databases are pinged and found to be standalone databases then JSM will remove the replica definition automatically when the last replica database is pinged).

Note that the replica definition is deleted, however the databases definitions will remain.



Remove a Database from a Group

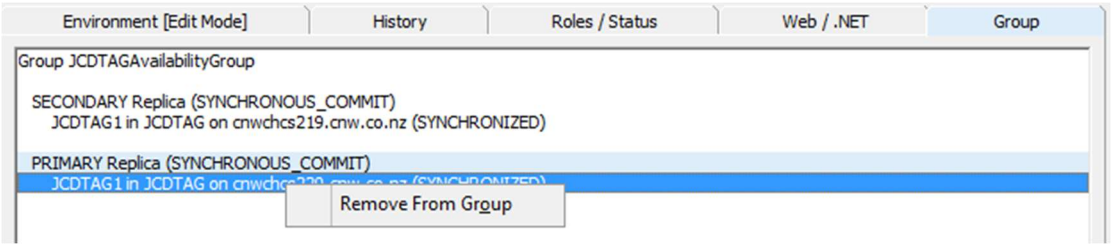
This option is provided so that a database can be removed from a group.

If a database is pinged and found to no longer belong to a group then JSM will remove it from the group automatically.

The operation is provided in the case that the database is not pinged after it is removed from a group using SQL Server Management Studio.

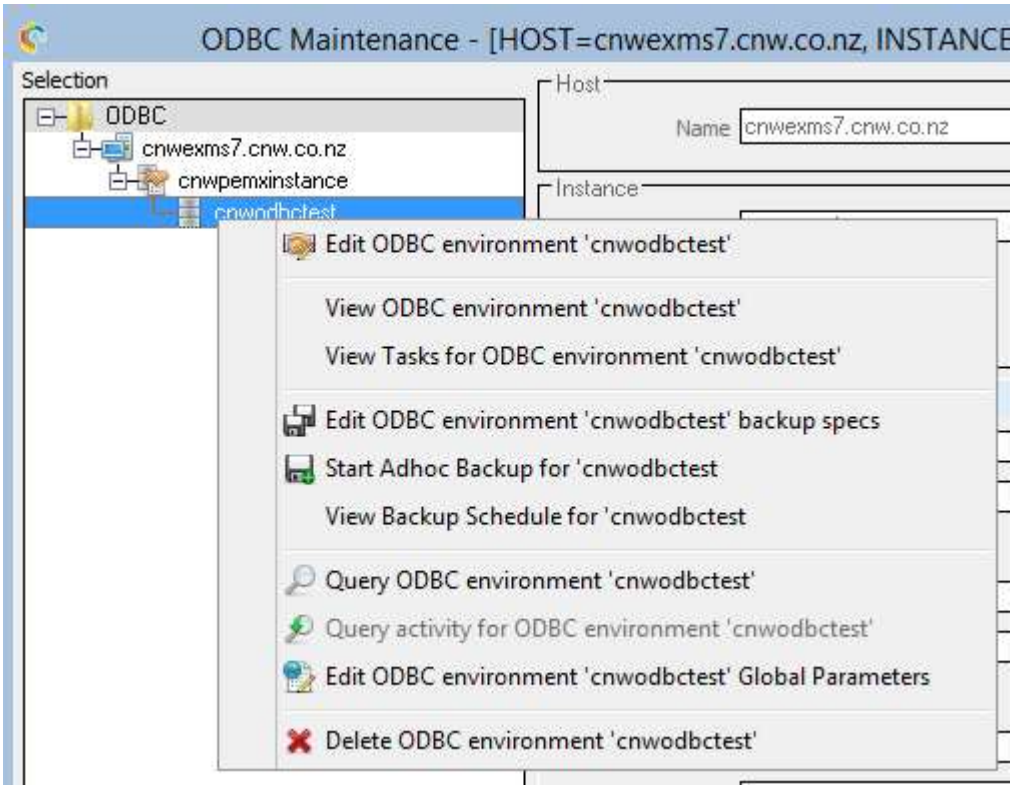
Note that this menu is only available when selecting a database in the Primary replica.

Note also that a database definition cannot be deleted in JSM when it is a member of a group (to ensure group definition consistency). When the last database is removed from a replica, the JSM group definition is removed.



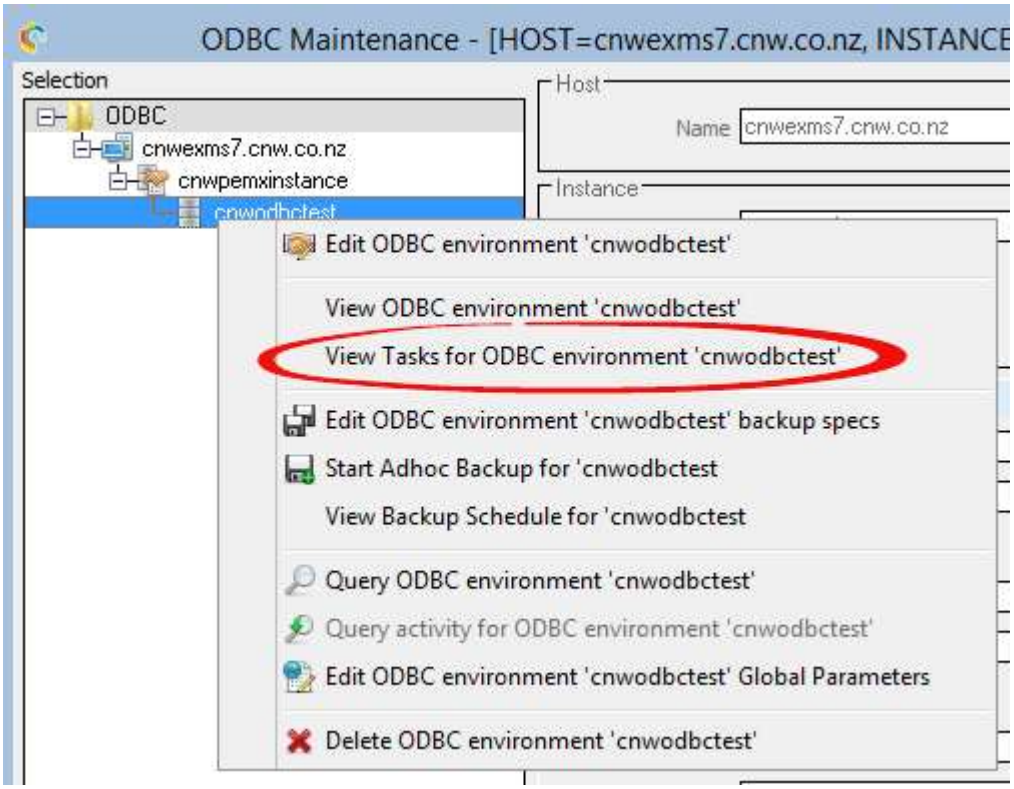
Display Tasks for an ODBC 'System' Environment

Select the appropriate 'managed' ODBC instance and right click to get the menu options, one of which is 'View Task Items for ODBC Instance <name>'



Display Tasks for an ODBC Environment

Select the appropriate 'managed' ODBC environment and right click to get the menu options, one of which is 'View Task Items for ODBC environment <name>'



JADE Systems Agent ODBC Log Settings

The log files produced by cnwsql.dll are prefixed cnwSql.log and are located in the same directory as standard Jade log file as defined in the JSA initialization file ([JadeLog] LogDirectory=<path>)

If, for some reason, this library fails to find a path, cnwsql.dll reverts to writing its logs in **c:\cardinal\logs**.

The level of logging in the cnwsql.log file can be modified by implementing a LogLevel attribute in the JSA initialization file by defining the following:

```
[OdbcMonitoring]
LogLevel=<value>
```

There are 4 levels of logging as follows:

LogLevel	Description
1	Only Error messages are written to the log file.
2 default	Level 1 + emSqlXXXX function exits/results
3	Level 2 + function entries/exits
4	Level 3 + other messages useful for debugging

JADE Systems Agent ODBC Applications

Once a new database has been defined and/or edited the configuration is sent to the designated JADE Systems Agents on each of the respective server hosts you have selected to be monitors.

On receipt of a new configuration or a configuration change, the **KarmaODBCChecker** application will start. An alert will be raised if the new or changed configuration fails to establish a successful connection to the intended ODBC database.

If **KarmaODBCChecker** completes successfully persistent data will be held in JSA identifying a successful check was completed. In addition, ODBC version information, database role and related status variables such as synchronization will be sent back to JSM. Providing no changes are made to database in the meantime, **KarmaODBCChecker** will only be run once.

Note: an initialised JSA will cause **KarmaODBCChecker** to run again.

KarmaODBCMonitor_<database name> will start once the configuration has been checked successfully and will run continuously until monitoring has been turned off. **KarmaODBCMonitor** queries the master database to determine the database role and various status variables (that are displayed in System Sub-View and other forms). If the database is a not a mirror then **KarmaODBCMonitor** attempts to connect to the ODBC database based on the user specified time interval.

Note: it may be advisable to tick 'Raise Alert' for each **KarmaODBCMonitor** application via the Host Environment Maintenance form 'Alarms & Exceptions' sheet for each JSA on each designated Server Host.

JADE Systems Agent ODBC Synchronization

When JSA first connects to JSM a synchronization process will take place to make sure the persistent data held in JSA is accurate and current. This is particularly

important for an initialised JSA as ODBC monitoring, if defined, should start without any user intervention being required.

This also prevents JSA from containing old configurations for changes that may have occurred while JSA was disconnected from JSM.

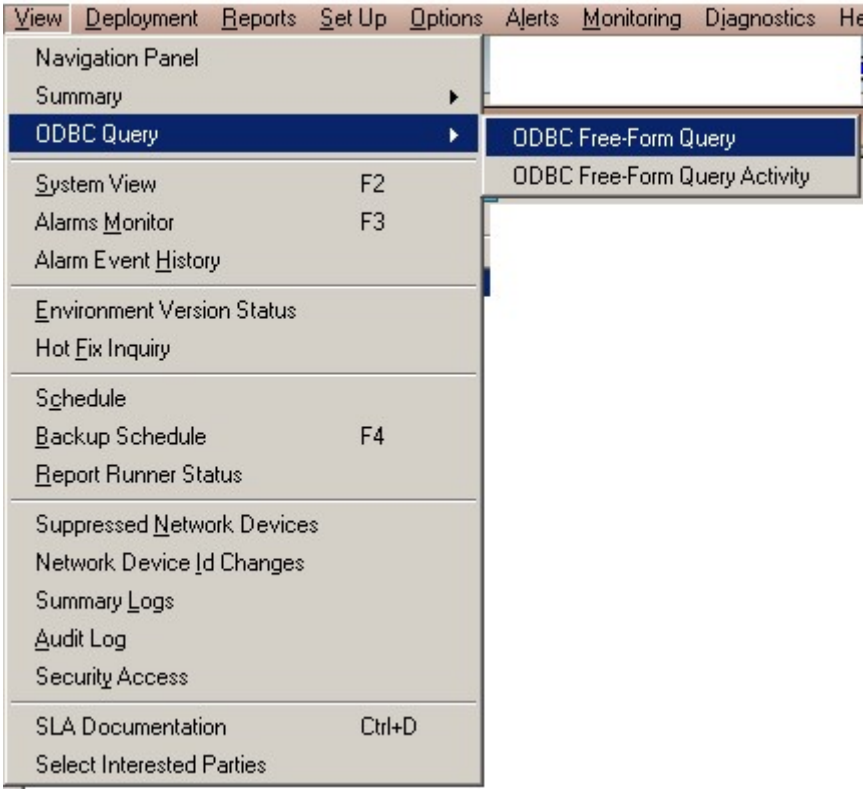
JADE Systems Agent Tanked Alerts

If JSA is unable to communicate with JSM then the last alert raised will be held persistently in JSA. When JSA is able to connect to JSM this tanked alert will be transferred.

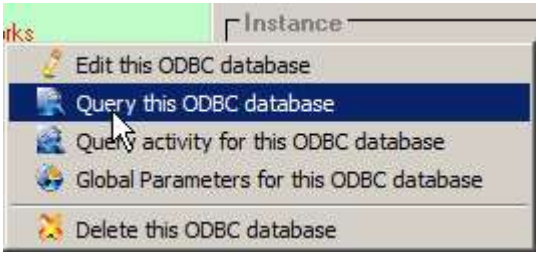
ODBC Free-Form Query

The 'ODBC Free-Form Query' form can be accessed from two places.

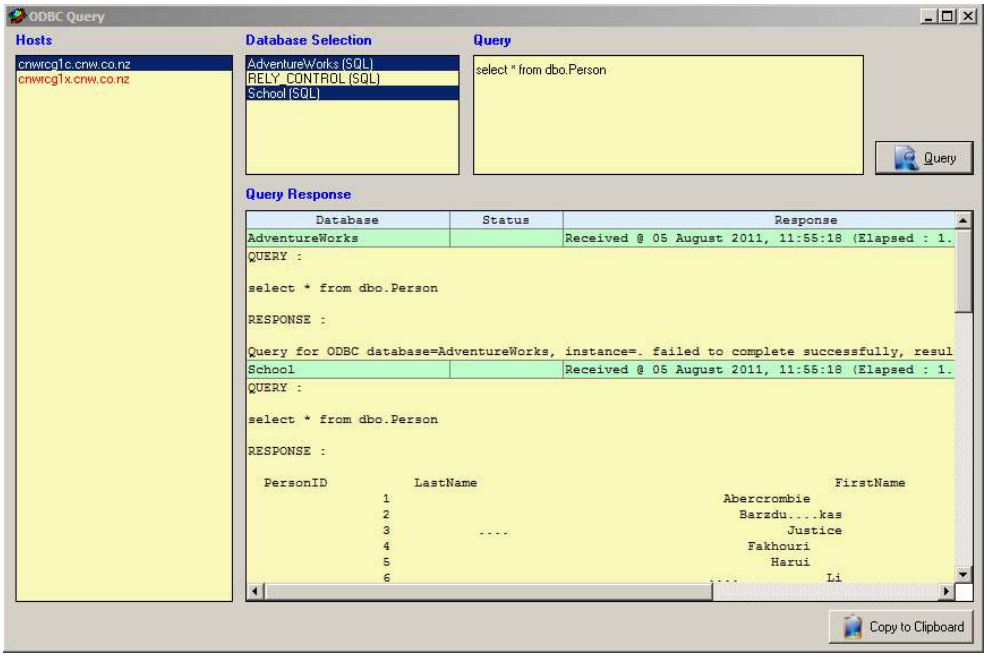
From the 'View / ODBC Free-Form Query' menu item:



or from the ODBC Maint form once a database definition has been selected.



Clicking one of these menu items will display the 'ODBC Query' form:



Any ServerHosts that have a JSA designated to monitor an ODBC database are shown in 'Hosts'. Hosts that are red denote an inactive or disconnected JSA at the time the 'ODBC Query' form was shown.

Once a ServerHost has been selected any databases this ServerHost is monitoring will populate 'Database Selection'. Query statements can be executed against multiple ODBC databases at the same time.

To perform a query, select the desired database entry (entries). Type your free-form query into the 'Query' field and click the 'Query' button (or simply press enter). Note that a new line in the 'Query' field is achieved while holding down the CTRL key and pressing ENTER.

Once the query statement has been submitted JSM will send the JSA on the selected ServerHost the query statement which will cause the **KarmaODBCQuery**_<database> name application to start and execute the query. There is one **KarmaODBCQuery** application for each of the databases selected. If the query is executed successfully then the response to that query is returned.

Returned responses are shown in the 'Query Response' table. You can copy the details of the response into your clipboard by clicking the cell in the table then clicking the 'Copy to Clipboard' button.

The 'ODBC Query' form is fully resizable and using it as a full screen form is the best option if your expecting a lot of data being returned in the response.

Queries are currently limited to 1000 characters.
The limits imposed by cnwsql are as follows:

result size	48,000 bytes
result rows	4096
result row size	4096 bytes
result column width	256 bytes (right truncated)

IMPORTANT

It is not JSM or JSA's responsibility to determine if your query statement is formatted correctly or if the items you're querying exist within the target ODBC database.

Being able to execute free-form queries puts a lot of power into the hands of the user, not only can the target database be read but it can also be changed and updated. Care must be taken when using the free-form query feature.

The username being used will have to have read access to the database and tables you wish to query, this is higher privilege than required by monitoring only.

ODBC Free-Form Query Activity

The ODBC Free-Form Query Activity form enables the operator to view a history of queries executed against ODBC databases. It includes information such as what query was executed, by whom and what the result was.

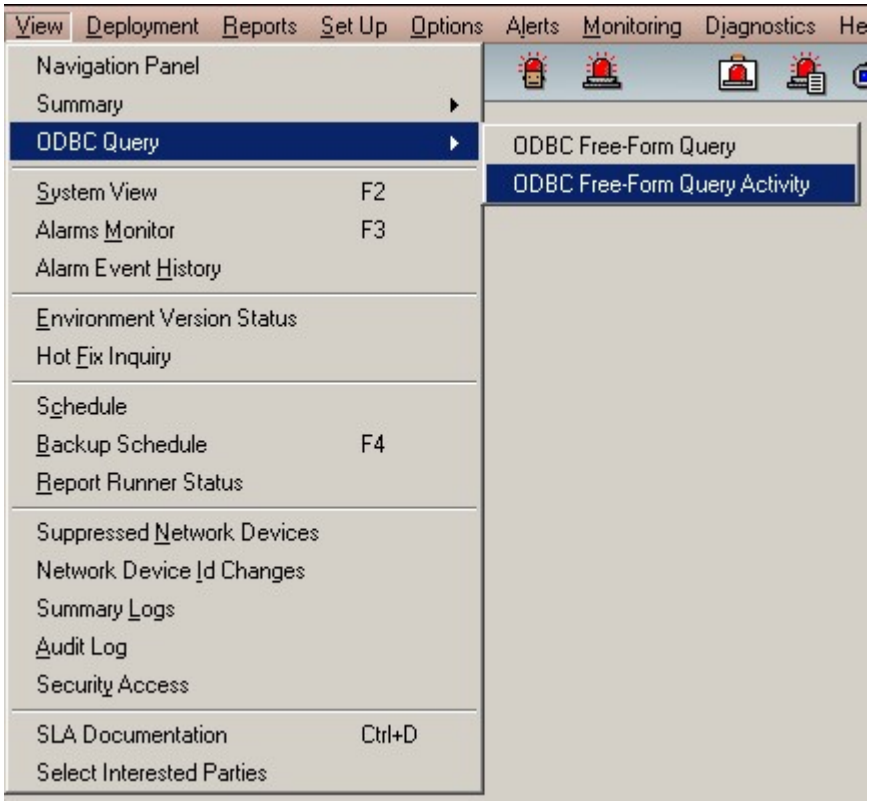
Because the ODBC Free-Form Query feature is so open and powerful we keep logs within JSM for each query executed. These logs include:

- the database the query was executed against
- who executed the query
- the time the query was executed
- the query statement used
- the response from the ODBC database with regards to that query

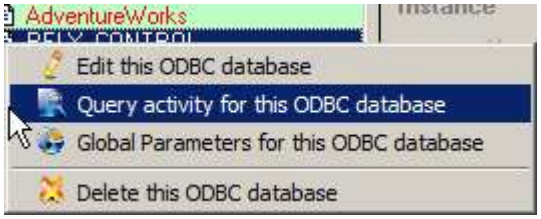
This logging is required for security reasons but also serves to provide a source of usable ODBC query statements and an indirect source of data held within and ODBC database.

ODBC query logs are retained from the date of the query plus the number of days defined in the 'System Global Options / Data Retention' 'Retention for miscellaneous logs' setting.

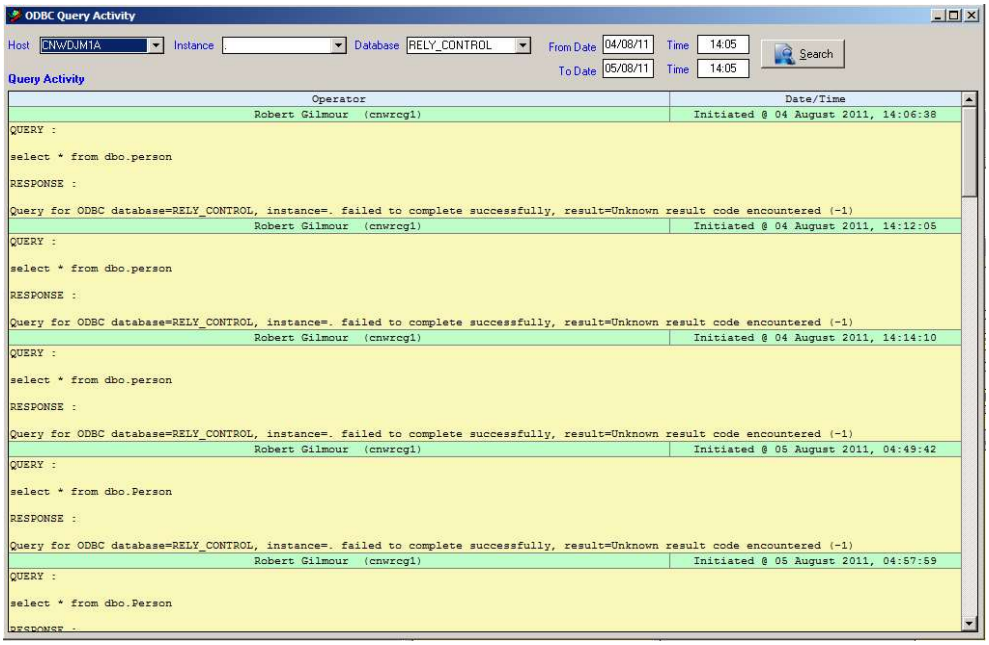
The 'ODBC Free-Form Query Activity' form can be accessed from two places. From the 'View / ODBC Free-Form Query Activity' menu item:



or from the ODBC Maint form once a database definition has been selected (Note: this menu item will be disabled if the selected database contains no query activity):



Clicking one of these menu items will present the 'ODBC Query Activity' form:

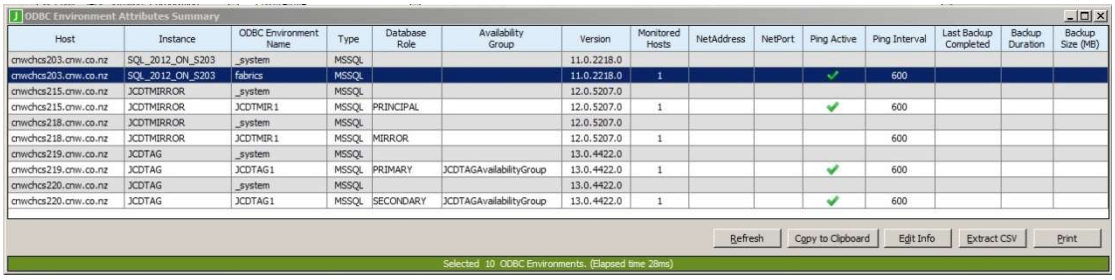


If this form has been displayed coming from the 'ODBC Maint' form then activity is automatically shown for the last 24 hour period. If this form is displayed via clicking the the 'View / ODBC Free-Form Query Activity' then the user must select their ODBC Host, Instance and Database. By default the 'From Date & Time' & 'To Date & Time' periods are setup to show activity for the last 24 hours.

Clicking the 'Search' button will display all ODBC query activity for the selected database within the user specified time frame.

ODBC Monitoring Summary

All defined ODBC monitored environments can be viewed on a single form. The **ODBC Environment Summary** form can be displayed by clicking the 'View / ODBC Environment Summary' menu item:



Host	Instance	ODBC Environment Name	Type	Database Role	Availability Group	Version	Monitored Hosts	NetAddress	NetPort	Ping Active	Ping Interval	Last Backup Completed	Backup Duration	Backup Size (MB)
cnwchcs203.cnw.co.nz	SQL_2012_ON_S203	_system	MSSQL			11.0.2218.0								
cnwchcs203.cnw.co.nz	SQL_2012_ON_S203	fabrics	MSSQL			11.0.2218.0	1			✔	600			
cnwchcs215.cnw.co.nz	JCDTMIRROR	_system	MSSQL			12.0.5207.0								
cnwchcs215.cnw.co.nz	JCDTMIRROR	JCDTMIR1	MSSQL	PRINCIPAL		12.0.5207.0	1			✔	600			
cnwchcs218.cnw.co.nz	JCDTMIRROR	_system	MSSQL			12.0.5207.0								
cnwchcs218.cnw.co.nz	JCDTMIRROR	JCDTMIR1	MSSQL	MIRROR		12.0.5207.0	1				600			
cnwchcs219.cnw.co.nz	JCDTAG	_system	MSSQL			13.0.4422.0								
cnwchcs219.cnw.co.nz	JCDTAG	JCDTAG1	MSSQL	PRIMARY	JCDTAGAvailabilityGroup	13.0.4422.0	1			✔	600			
cnwchcs220.cnw.co.nz	JCDTAG	_system	MSSQL			13.0.4422.0								
cnwchcs220.cnw.co.nz	JCDTAG	JCDTAG1	MSSQL	SECONDARY	JCDTAGAvailabilityGroup	13.0.4422.0	1			✔	600			

Double clicking on an entry will take you to the ODBC configuration for that entry.

.NET Services and Web Info Discovery.

This feature was implemented in release 3.8.03.

Its intent is to 'discover' .NET Services and IIS Web site information from JADECare managed servers, to correlate the information in JSM to servers and managed environments (JADE and ODBC) and to provide the information via a number of forms. The details are expected to be used in subsequent releases in global parameter files, in deployments and in various auditing functions.

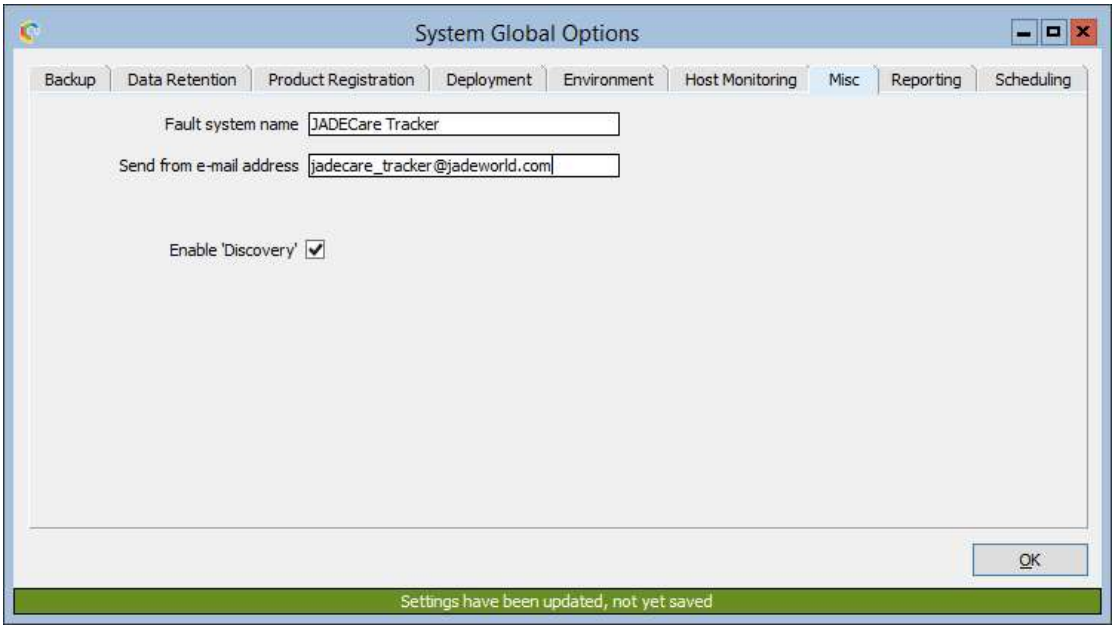
Turning the feature On/Off

The feature can be turned on or off for all JADECare managed servers, if the feature is on globally, then individual servers can specify whether or not the 'discovery' feature is to occur on them.

Turning the feature On/Off globally

The 'discovery' feature can be turned on (or off) globally for all servers managed by JSM via the **System Global Options** form.

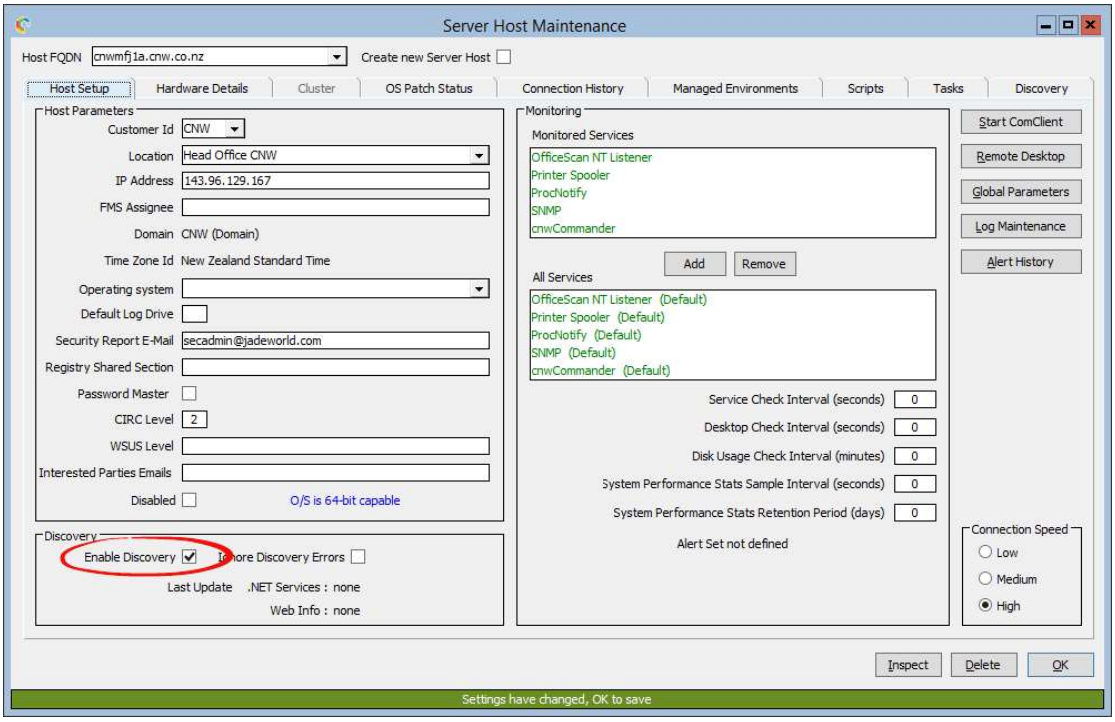
On the tab Misc, there is a check box captioned 'Enable 'Discovery'. Ticking the check box and committing the change means that the feature is now enabled globally and various controls/columns on various forms will now be visible.



Note that if the feature is turned off globally, that some .NET Services and/or IIS Web site information has already been discovered and these entities may need to be physically deleted if not required.

Turning the feature On/Off by Server

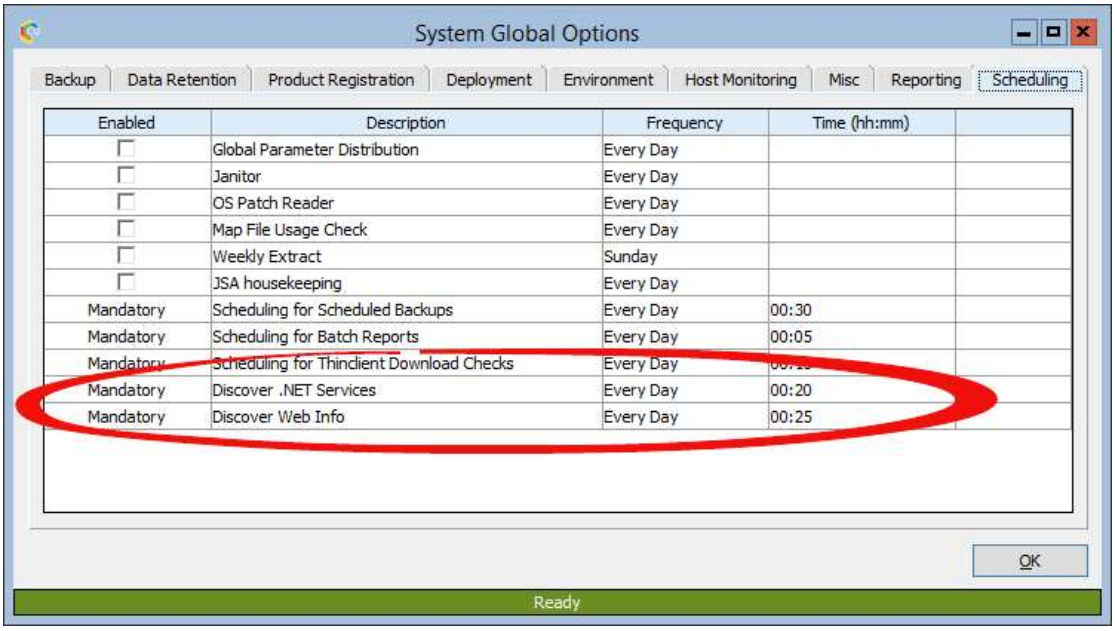
Once the discovery feature has been enabled globally, then each server can specifically have the discovery feature enabled or disabled. The default is disabled, this means that the feature can be gradually rolled out across all the servers required.



Scheduling the Discovery Process

Once the feature has been turned on globally then it is expected that the discovery processes will be scheduled daily. This is done on the **System Global Options** form in the Scheduling tab. Note that the default times are:

- Discover .NET Services – 00:20
- Discover Web Info – 00:25



The scheduled item 'Discover .NET Services' will cause JSA application **KarmaDotNetServicesDiscovery** to be started at the scheduled time and will obtain all .NET Service details for the server. If any changes from the previous .NET Services discovery, the details will be forwarded to JSM to maintain .NET Service entities and related entities within JSM.

The scheduled item 'Discover WebInfo' will cause JSA application **KarmaWebInfoDiscovery** to be started at the scheduled time and will obtain all IIS Web site details for the server. If any changes from the previous .NET Services discovery, the details will be forwarded to JSM to maintain web entities and related entities within JSM.

Ad-Hoc Discovery Requests

If the feature has been turned on globally and for a server, then the System View form provides two extra menu options when right clicking on a Server entity, being:

- Discover .NET Services
- Discover Web Info

Clicking on the 'Discover .NET Services' will cause JSA to execute application **KarmaDotNetServicesDiscovery** immediately and will obtain all .NET Service details for the server. If any changes from the previous .NET Services discovery, the details will be forwarded to JSM to maintain .NET Service entities and related entities within JSM.

Clicking on the 'Discover Web Info' will cause JSA to execute application **KarmaWebInfoDiscovery** immediately and will obtain all IIS Web site details for the server. If any changes from the previous IIS Web info discovery, the details will be forwarded to JSM to maintain web entities and related entities within JSM.

Use of JadeCare.cfg File

Both .Net Services and WebAppPools use the JadeCare.cfg file to provide a direct reference to a JADECare Managed Environment. In the case of .Net Services the use of a JadeCare.cfg file is mandatory and failure to provide the required definition will result in the .Net Service being ignored while raising an alert or Tracker.

The location of the JadeCare.cfg is defined :

- By deriving the 'misc' node of the .Net Services path
- By using the physical path from the first Web Application defined for a WebAppPool and deriving the 'misc' node.

The path used not only derives the 'misc' path but also the type of Managed Environment expected:

- If the node is 'c_misc' the Managed Environment type is JADE
- If the node is 'misc' the Managed Environment type is ODBC

NOTE: If a WebAppPool's derived physical path does not have 'misc' node then no attempt is made to find a JadeCare.cfg and no advisory is issued as this is not considered an error situation.

The optional key 'SuppressAdvisory=<true|false>' will suppress the generation of Traker calls / alerts specifically for the .Net Service / Web Site Application for which the option is defined. This option does not suppress configuration / setup issues; for example jadehttp.ini definition being incomplete.

.Net Services Only

In some cases we can also determine the database identity and server name for the managed environment from the executable path name, see below. The use of a JadeCare.cfg file can override the derivation of Instance Name, Environment Name and Server/Cluster and may be required to define the missing values.

- If the executable path contains '\misc\apps\' and is preceded by two directory nodes, then the .NET Service is deemed to be running on the database server and the first directory node defines the ODBC Instance Name and the second directory node defines the ODBC Environment Name
- If the executable path contains '\misc\apps\' and is preceded by one directory node, then the .NET Service is deemed to be running on a different server from the ODBC database, which is unknown, and only the ODBC Environment Name can be derived from the first directory node.

  In this case a JadeCare.cfg file is mandatory to specify, at minimum, the ODBC Instance Name and the ODBC Server.
- If the executable path contains '\server\c_misc\apps\' and is preceded by one directory node, then this .NET Service is executing on the same server as the JADE database managed environment (it's using the same directory structure) and the JADE Environment Name is derived from the first directory node.
- If the executable path contains '\c_misc\apps\' and is preceded by one directory node, then this .NET Service is executing on a different server from the JADE database managed environment but the JADE Environment Name is derived from the first directory node.

  In this case a JadeCare.cfg file is mandatory to specify, at minimum, the JADE Database Server.

The 'logPath' is now either derived from the JadeCare.cfg file or from scanning all local volumes for a log path derived from the executable path.

If the logpath is defined in the JadeCare.cfg file, it is validated to check that it is a valid path name and that it exists. Failure in either of these two rules will generate an advisory message. Note that the JadeCare.cfg definition will be used if it is valid and exists regardless of directory discovery.

The log path that is derived from the executable path is scanned for all volumes (not mapped drives).

- If no match is found and the config file has not defined the logpath then an advisory is generated
- If multiple entries are found an advisory is generated
- If the config file definition does not match the entry located an advisory is generated

If the JadeCare.cfg file does NOT define the logPath, then if a single logPath is found that is the value returned.

The JadeCare.cfg definition for the log path (if required) is:

```
[<.Net Service name>]
LogPath=<full path name>
```

Example:

If the executable path is 'r:\example\server\c_misc\DataDump\service.exe' then the base logpath will be ':\example\logs' and this directory will be checked on all volumes local to the server.

If 'r:\example\logs' is:

- located and
- is the only directory located in the scan and
- the JadeCare.cfg did not define this value or the JadeCare.cfg defined the same value, then the logpath returned to JSM will be 'r:\example\logs'

JadeCare.cfg File

This file is expected to reside in the miscellaneous directory (\misc for ODBC environments and \c_misc for JADE environments).

For .Net Services the content of this file can be used to provide the items that in some cases cannot be derived from the executable path but can also be used to explicitly override any or all of the values derived from the executable path.

For WebAppPools, if one or more of the expected entries is not defined then an alert or Tracker is raised and the JadeCare.cfg content is ignored for the WebAppPool. In that case the associated Managed Environment will be determined from the first part of the WebAppPool name (delimited by underscore (_)).

The file is defined in ini file format and may be used by more than one .NET Service and/or WebAppPool where more than one .NET Service / WebAppPool is defined for a managed environment. The section name is either the full .Net Service name or the full WebAppPool name.

```
[<service name 1>]
InstanceName=<ODBC Instance Name>
EnvironmentName=<ODBC or JADE Environment Name>
ServerOrClusterName=<server or cluster name>
LogPath=<full path name>       // this field is optional
SuppressAdvisory=<true|false>  // this field is optional


[<service name 2>]
EnvironmentName=<ODBC or JADE Environment Name>
ServerOrClusterName=<server or cluster name>

[<service name n>]
InstanceName=<ODBC Instance Name>
EnvironmentName=<ODBC or JADE Environment Name>
ServerOrClusterName=<server or cluster name>

[<webapppool name 1>]
InstanceName=<ODBC Instance Name>
EnvironmentName=<ODBC or JADE Environment Name>
ServerOrClusterName=<server or cluster name>

[<webapppool name 2>]
EnvironmentName=<ODBC or JADE Environment Name>
ServerOrClusterName=<server or cluster name>

[<webapppool name n>]
InstanceName=<ODBC Instance Name>
EnvironmentName=<ODBC or JADE Environment Name>
ServerOrClusterName=<server or cluster name>
```

.NET Services Discovery

Each Windows Service that the discovery process is interested in will have an executable path name that contains either '\misc\apps' or '\c_misc\apps. These services are to be defined as .NET Services.

Any of the Windows Services may also be clustered in which case the cluster information is also passed back to JSM.

Any Service with a path name that does not conform to that standard will be ignored.

.NET Service Discovery Errors

If the required items cannot be derived and also cannot be derived from a JadeCare.cfg file, then:

- the details for this .NET Service will be dropped from the discovery text buffer sent to JSM
- if the 'Ignore Discovery Errors' is NOT set for this server then an error advice with appropriate details will be issued which will create a Tracker call (if Tracker instantiated) or create an alert.

Examples:

Example 1 - .NET Service on same server as ODBC Database.

Discovery is on server **cnwexms1**, Windows Service name is **cnwexample** and executable path is 'e:\cnwinstance\cnwexample\misc\apps\datareader\ datareaderservice.exe'.

As the path contains literal '\misc\apps\' then this is considered a .NET Service and the managed environment is an ODBC Environment.

The \misc\apps\ sub-directory is preceded by two directory nodes so the ODBC Instance name can be derived as can the ODBC Environment Name, in this case they are 'cnwinstance' and 'cnwexample' respectively.

Further, as the literal '\misc\apps\' is preceded by two directory nodes then the .NET Service is deemed to be executing on the same server as the ODBC managed environment, that is **cnwexms1**.

In this example, the JadeCare.cfg is not required as all three required items have been derived, that is:

- ODBC Instance name
- ODBC Environment name
- Server or Cluster name

That does not preclude the JadeCare.cfg from being defined and overriding one, two or all three of the derived values. If defined, then the JadeCare.cfg would be expected in directory 'e:\cnwinstance\cnwexample\misc' with following entries

```
[cnwexample]
InstanceName=cnwinstance
EnvironmentName=cnwexample
ServerOrClusterName=cnwexms1
```

Example 2 - .NET Service on different server from ODBC Database.

Discovery is on server **cnwexms1**, Windows Service name is **cnwexample** and executable path is 'e:\ cnwexample\misc\apps\datareader\ datareaderservice.exe'.

As the path contains literal '\misc\apps\' then this is considered a .NET Service and the managed environment is an ODBC Environment.

The \misc\apps\ sub-directory is preceded by one directory node so the ODBC Instance name and server that the target ODBC environment is on cannot be derived. The ODBC Environment Name can be derived and is 'cnwexample'.

In this example, the JadeCare.cfg is required as only one of the three required items can be derived, that is ODBC Environment name.

The JadeCare.cfg is expected in directory 'e:\cnwexample\misc' with following entries, the minimum required entries being InstanceName and ServerOrClusterName

```
[cnwexample]
InstanceName=cnwinstance
EnvironmentName=cnwexample
ServerOrClusterName=cnwexms1
```

Example 3 - .NET Service on same server as JADE Database.

Discovery is on server **cnwexms1**, Windows Service name is **cnwexample** and executable path is 'e:\cnwexampl\server\c_misc\apps\datareader\ datareaderservice.exe'.

As the path contains literal '\c_misc\apps\' then this is considered a .NET Service and the managed environment is a JADE Environment.

The \c_misc\apps\ sub-directory is preceded by two directory nodes with second directory node being '\server\' then the first directory node is used to derive the JADE Environment name and the .Net service is deemed to be executing on the same server as the JADE database environment. In this case the JADE Environment name is 'cnwexampl' and the JADE Environment server is "cnwexms1'

In this example, the JadeCare.cfg is not required as the two required items have been derived, that is:

- JADE Environment name
- Server or Cluster name

That does not preclude the JadeCare.cfg from being defined and overriding one or both of the derived values. If defined, then the JadeCare.cfg would be expected in directory 'e:\\cnwexampl\server\c_misc' with following entries

```
[cnwexample]
EnvironmentName=cnwexampl
ServerOrClusterName=cnwexms1
```

Example 4 - .NET Service on different server from JADE Database.

Discovery is on server **cnwexms1**, Windows Service name is **cnwexample** and executable path is 'e:\ cnwexampl\c_misc\apps\datareader\ datareaderservice.exe'.

As the path contains literal '\c_misc\apps\' then this is considered a .NET Service and the managed environment is a JADE Environment.

As the \c_misc\apps\ sub-directory is NOT preceded by directory node '\server\' then the .NET Service is executing on a different server from the JADE database environment. The JADE Environment Name can be derived from the first directory node and is 'cnwexampl'.

In this example, the JadeCare.cfg is required as only one of the two required items can be derived, that is JADE Environment name.

The JadeCare.cfg is expected in directory 'e:\cnwexampl\c_misc' with following entries, the minimum required entry being ServerOrClusterName

```
[cnwexample]
EnvironmentName=cnwexampl
ServerOrClusterName=cnwexms1
```

IIS / Web Details Discovery

All websites, web applications and web app pools will be parsed and details provided to JSM.

The following prefixed App Pool names are deemed to be 'default' app pool names:

- default
- asp.net
- classic
- apppool

Automatic Association With Managed Environment

Every App Pool, except for the abovedefault ones, will try and make an automatic association with a managed environment. This association can be implicit or explicit.

Explicit Association:

Explicit association with a managed environment can be achieved by specifying the App Pool name as a section name within a JadeCare.cfg file (see above) and providing the required parameters within that section that will uniquely identify a Managed Environment (JADE or ODBC).

The association will default to using implicit association under the following circumstances:

- the physical path does not contain a 'misc' node (either \misc or \c_misc)
- the 'misc' path derived does not contain a JadeCare.cfg file
- the JadeCare.cfg file does not contain the App Pool section name
- the JadeCare.cfg file does not contain all the parameters required to uniquely define the Managed Environment

Implicit Association:

Implicit association will occur if the discovery details are not extended by information gleaned from a JadeCare.cfg file.

Implicit association will use the first part of the App Pool name delimited by an underscore (_) to determine the managed environment name.

If the App Pool name does not contain an underscore then the App Pool is NOT consider to be one that can be associate with a Managed Environment and no automatic association will be attempted.

Using the derived managed environment name JSM searches all managed environments within JSM for a match on name, this includes JADE database environments, JADE remote environments and ODBC environments, the search being case-insensitive. When a single unique match is found then an automatic association is made. When no match is made or more than one match is made, then an advisory is issued advising of that.

JadeHttp Details and Extensions

Where a web application has a jadehttp.dll associated with it, the jadehttp.dll version is 'discovered'.

Also the jadehttp.ini file is expected to be located in an associate in sub-directory (derived as <physical path>_http\ini\jadehttp.ini) and each section that defines a TcpConnection key (which may also be appended with a number) also expects to locate the following :

- EnvironmentName
- ServerOrClusterName

Failure to identify these items will cause an advisory to be generated.

These two items are required to allow the appropriate JADE Environment to be associated with the WebJadeHttp entities and be viewable via various GUO forms.

WebSite Bindings and Certificates

A website can have one or more bindings and the http/https binding can also define a certificate. These details and the certificate expiry are retrieved so as to be visible from JSM.

Known Restriction:

It is possible to get into a situation where:

- the certificate appears to have been removed from the certificate store and
- for the web site binding that previously defined the deleted certificate to appear to not define a certificate.

However if the following command is executed (in a Command dialog) then it shows that the web site binding is still pointing to the certificate. And JSM discovery will still show the certificate and web site binding using that certificate.

```
netsh http show sslcert
```

In order to avoid this possible issue, do the following in order:

- delete the affected web site binding
- if still required, add the web site binding without the certificate or with a new certificate.
- delete the certificate
- let discovery occur to update JSM

Diagnostic Facilities

Refer to [JadeExecuteFlags] ini file section options:

- Joss_Discovery_DisplayFullBuffer
- Joss_Discovery_IgnoreLastBuffer

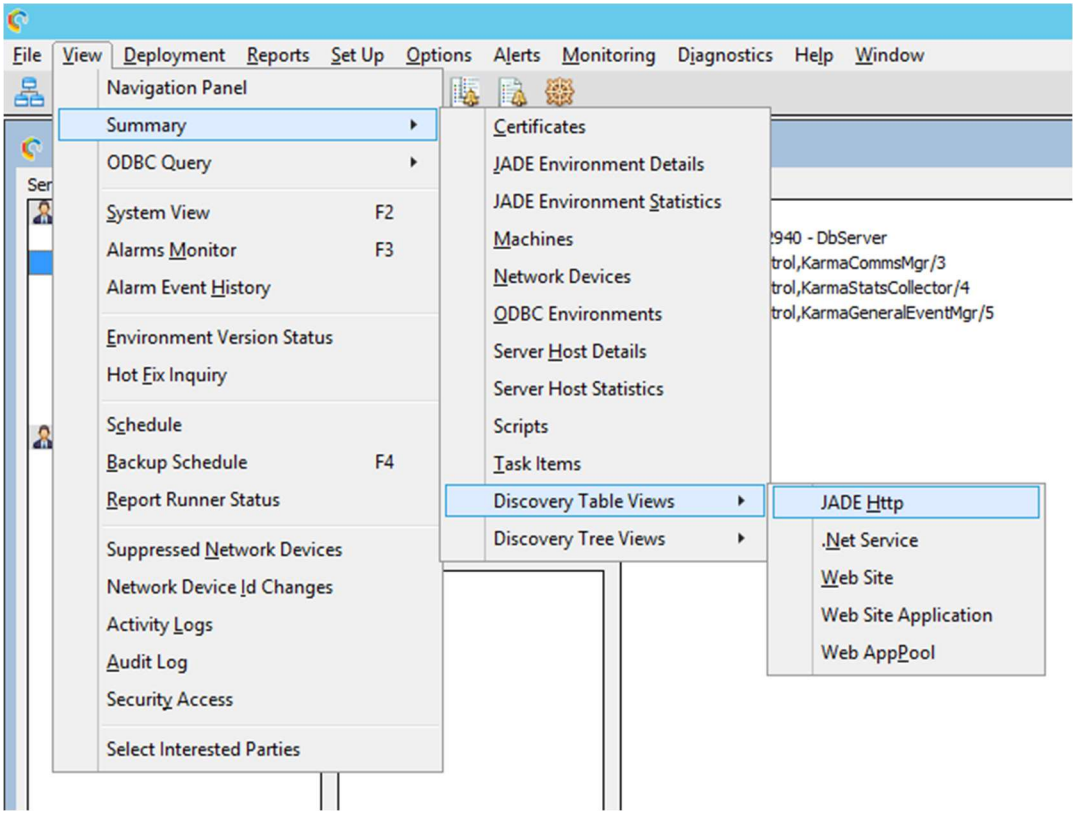
New Forms and Form Changes

When web discovery is enabled,

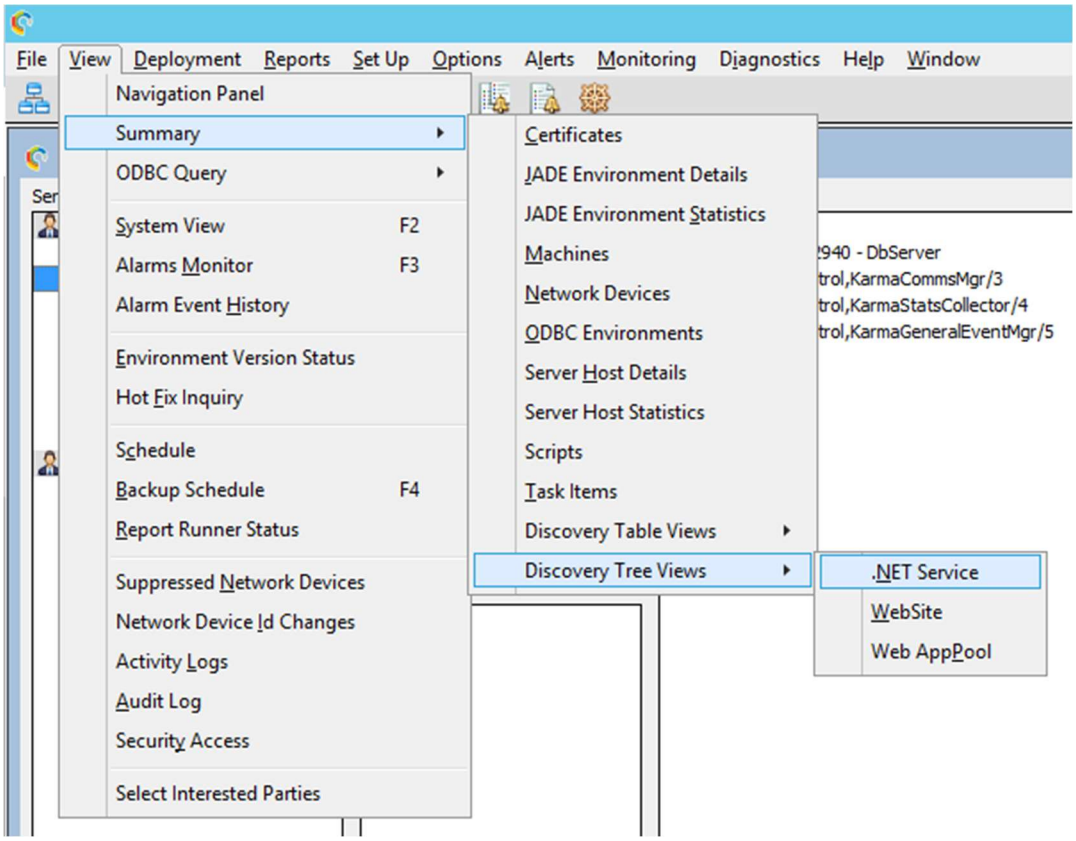
- New table view summary forms will be available in 'View / Summary /Discovery Table Views'

- New tree view summary forms will be available in 'View / Summary / Discovery Tree Views'.
- Also 'View / Summary / Certificate Details' will contain Web site SSL certificate information.

Discovery Table Views

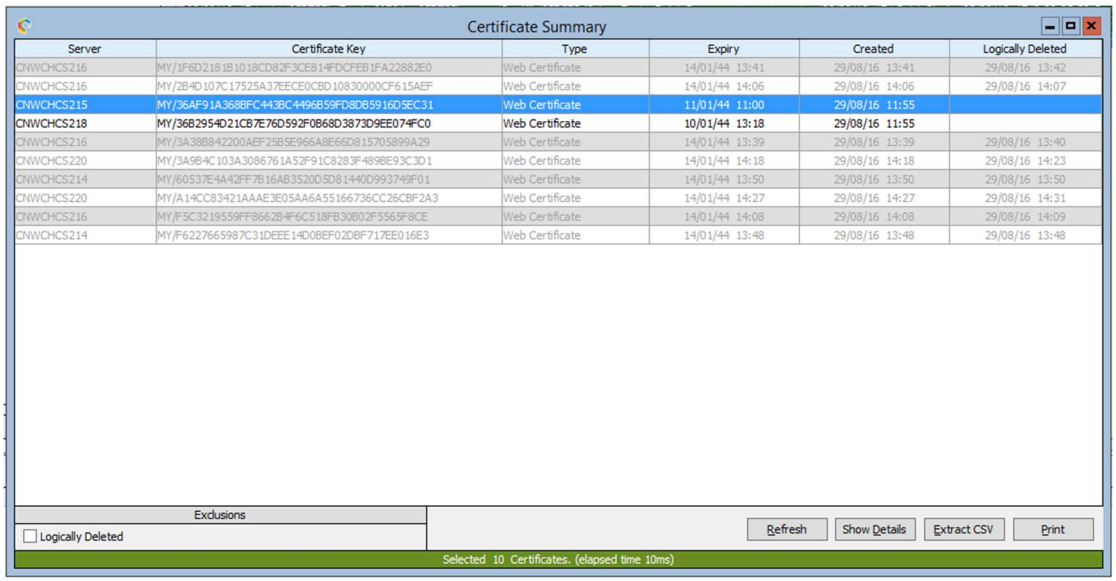


Discovery Tree Views

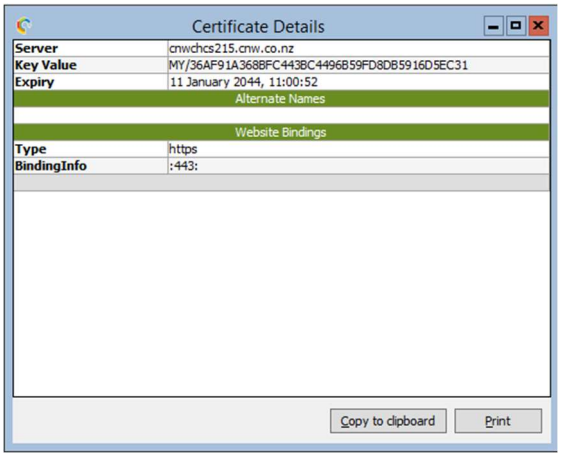


Certificates will display a summary of discovered certificates (Web SSL as well as application server SSL certificates and SSL certificate authority files).

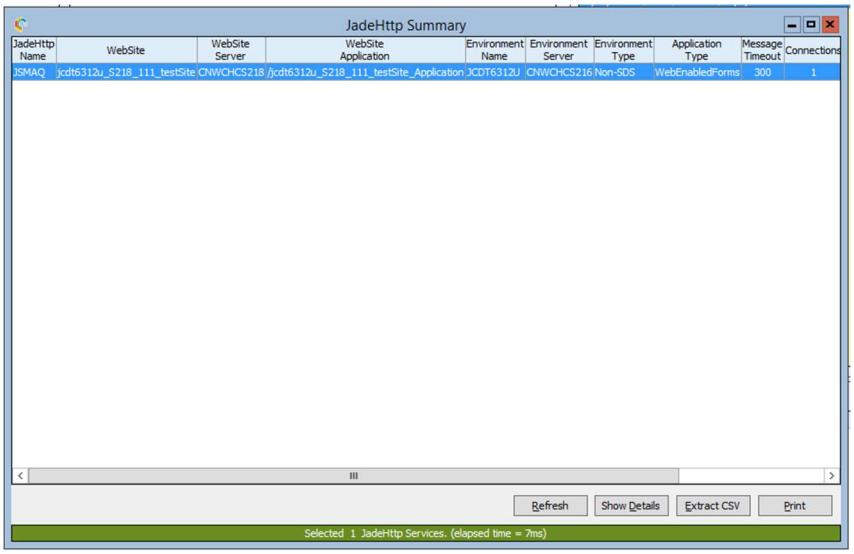
- This displays some brief information about the certificate, probably the most important being the expiry date.
- If 'Expiry' date cell for a particular certificate is red means that certificate is about to expire within 28 days.
- Logically deleted items will be greyed out and only visible when user uncheck 'Logically Deleted' exclusion (by default it will be checked).
    - This functionality is same for **Web Site**, **Web Site Application** and **Web AppPool** table view summary forms.



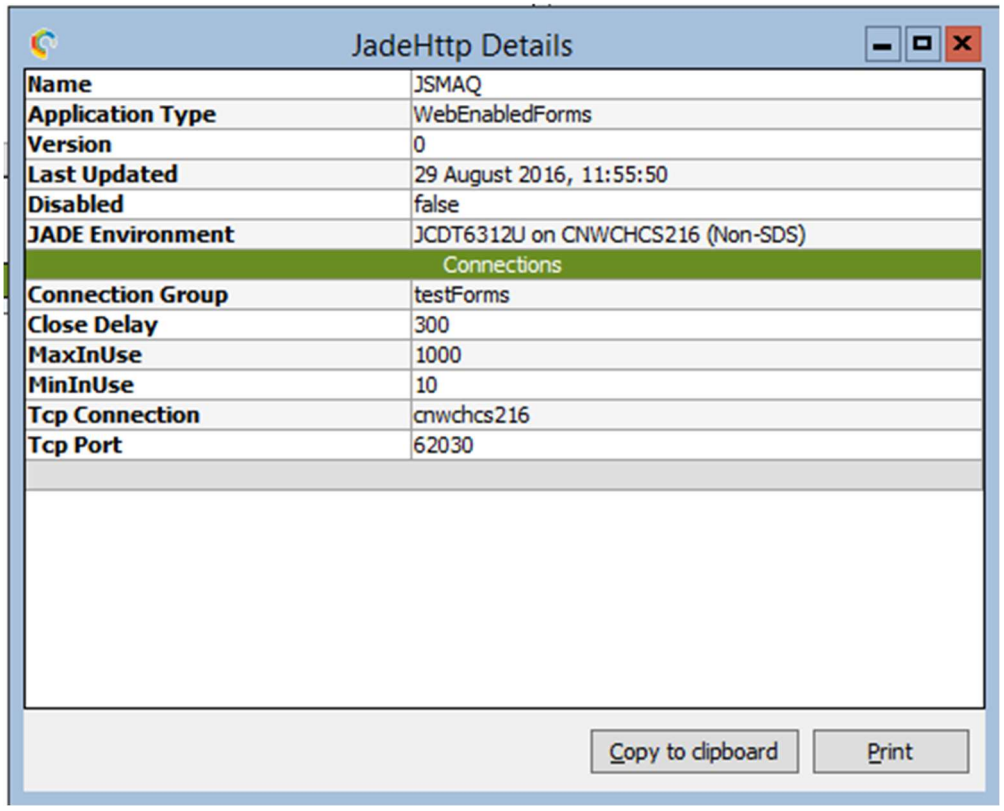
To get more information about selected certificate use 'Show Details' button or double click on it.



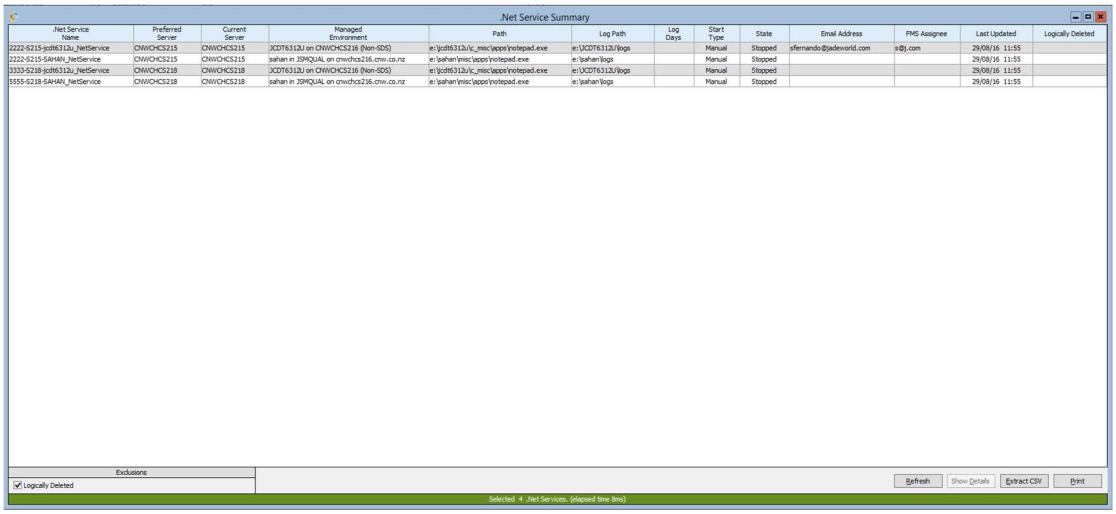
JADE Http displays a summary of web applications using JadeHttp.



To get more information about selected JADE Http site use ‘Show Details’ button or double click on it.



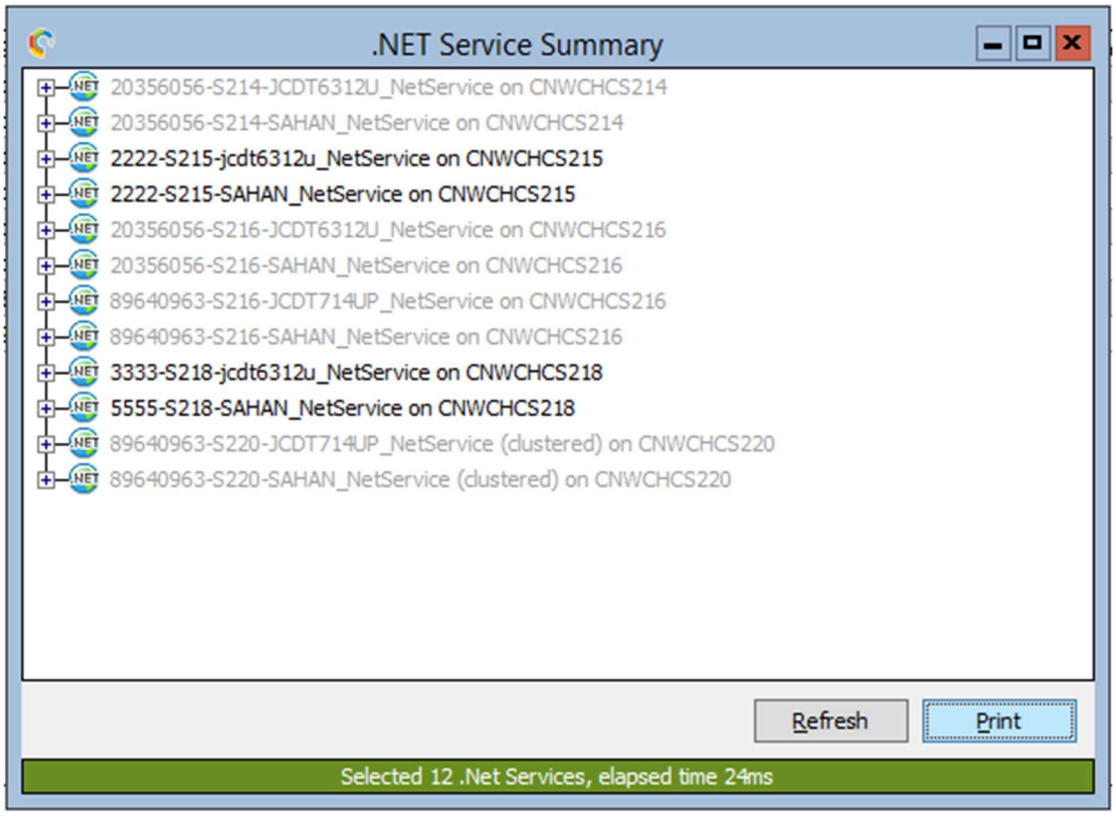
.**NET Service** table view summary will display all discovered .NET services



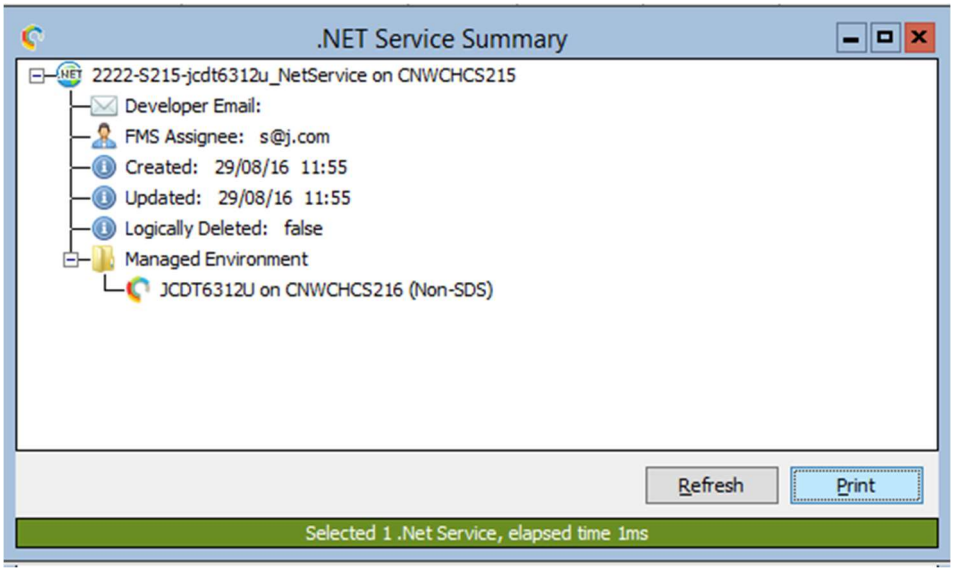
- To get more information in table view form about selected item use 'Show Details' button or double click on it.
    - It will show selected item information in tree view form.
    - This functionality is same for **Web Site**, **Web Site Application** and **Website AppPool Website** and **Website AppPool** summary forms **Website** and **Website AppPool** table view summary forms

.**NET Service** tree view summary

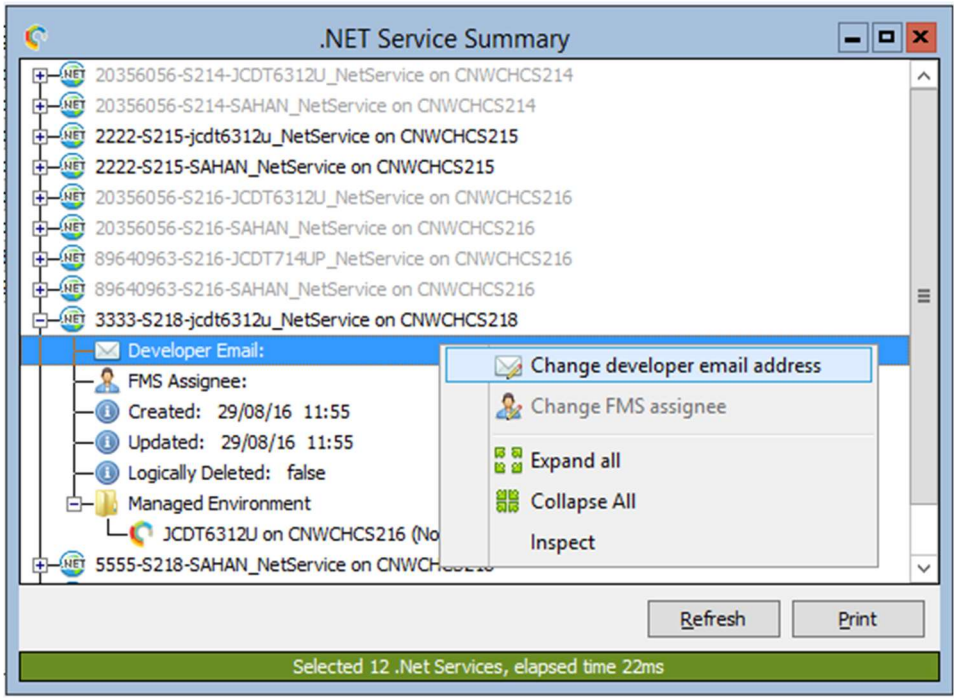
- Logically deleted items will be greyed out.
    - This functionality is same for **Web Site** and **Web AppPool** tree view summary forms.



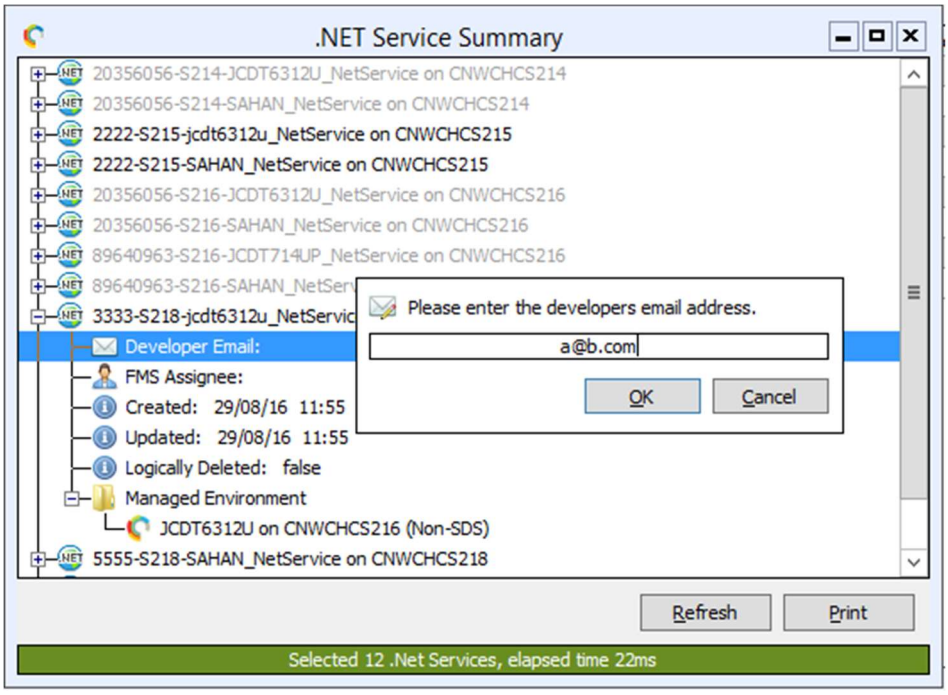
- To get more information expand selected item.
- Clustered .NET environments will include 'Clustered' text at the end of environment host name.
    - It is same for **Website** and **Website AppPool** tree view summary forms**.**



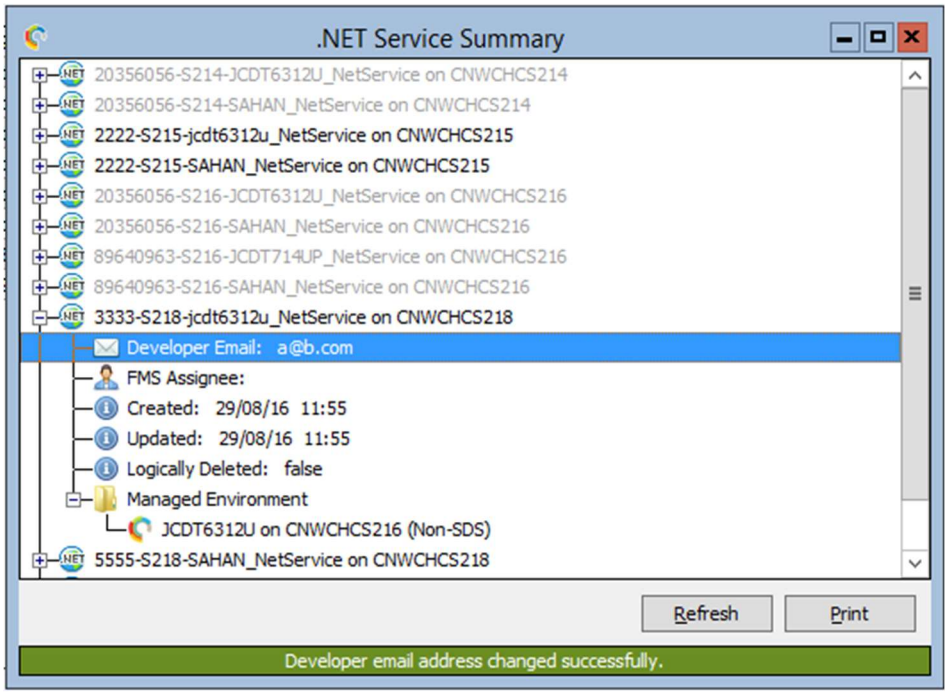
'Developer Email' and 'FMS Assignee' are editable entities on this form. To edit these items you simply select the entry and use the popup menu item



Once you click you will be presented with an input box. Note that the input box will always be presented directly beside the selected entry.



Committing this change is reflected in the selected entry

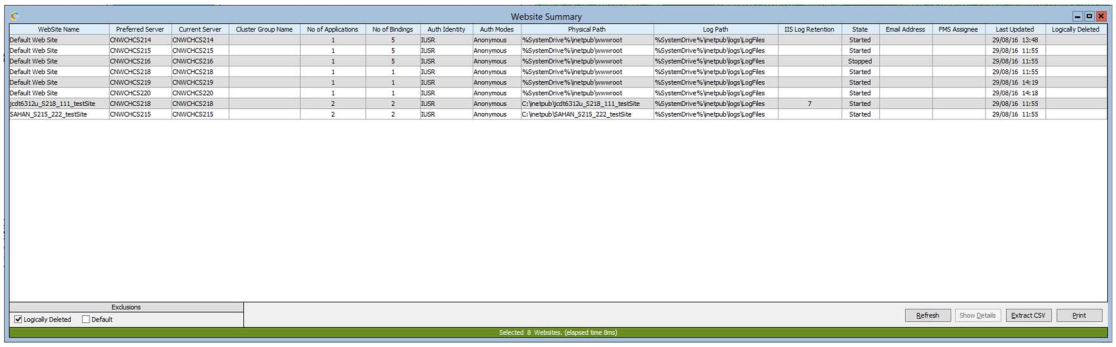


The process above also applies to **Website** and **Website AppPool** summary forms**.** 'Expand All' and 'Collapse All' are common across the same tree view forms and are used to expand or collapse the entire tree.

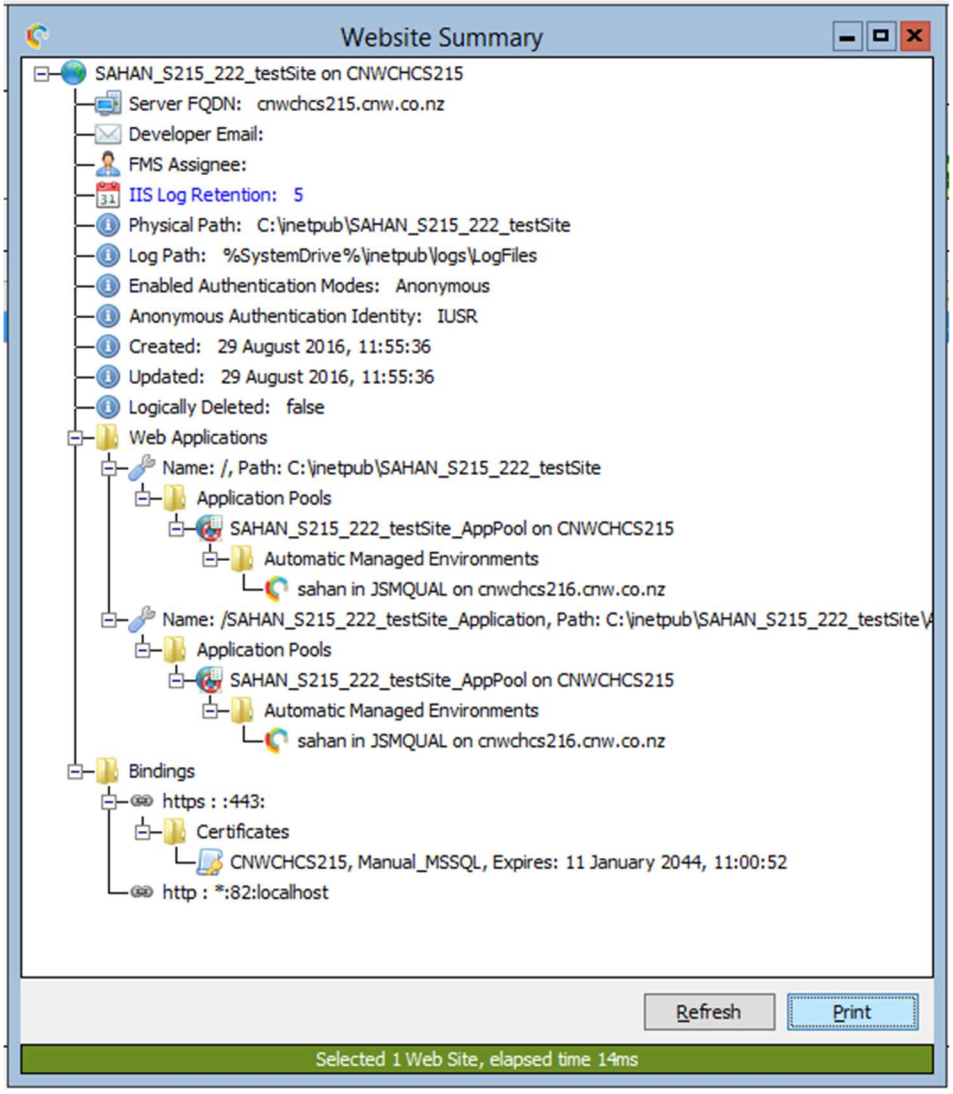
Selecting an entry and hitting Control + C on your keyboard will copy the tree into your clipboard. Clicking the 'Print' button will print a WYSIWYG report of the tree.

Web Site table view summary view.

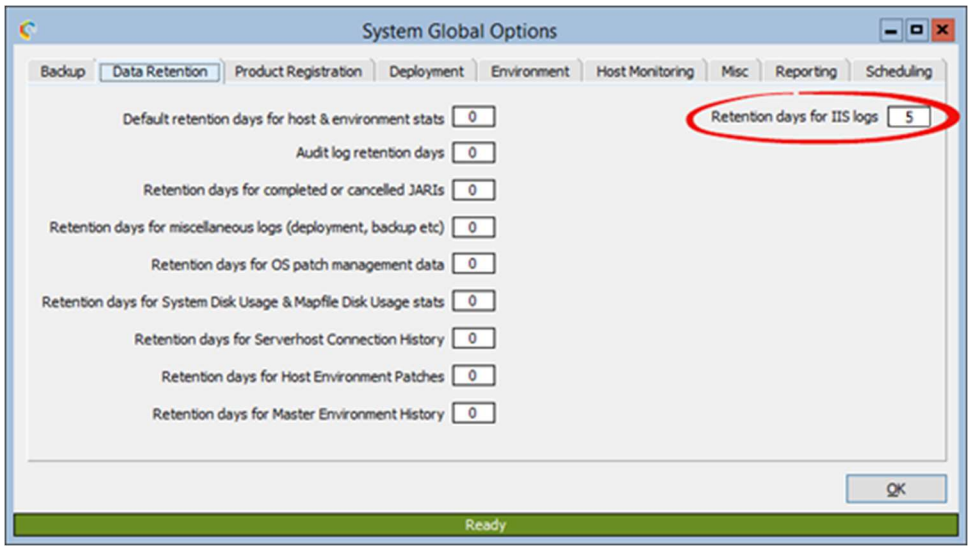
- Default items can be seen when 'Defaults' exclusion is unchecked (by default it will be checked).
    - This functionality is same for W**eb Site Application** and **Web AppPool** table view summary forms.



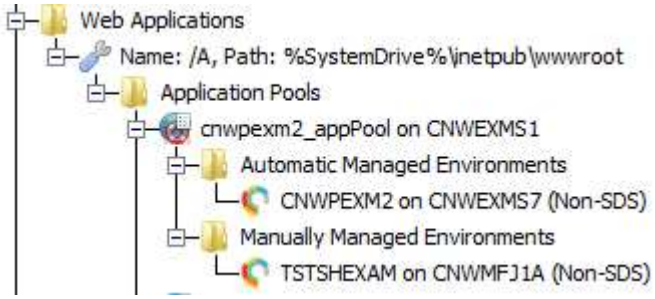
Web Site tree view summary provides a complete overview of a website.



In addition to 'Developer Email' and 'FMS Assignee' you can edit 'IIS Log Retention'. If the retention has not been explicitly declared for this website and the value is being inherited from the value defined in **Global Options** it will be displayed in blue.

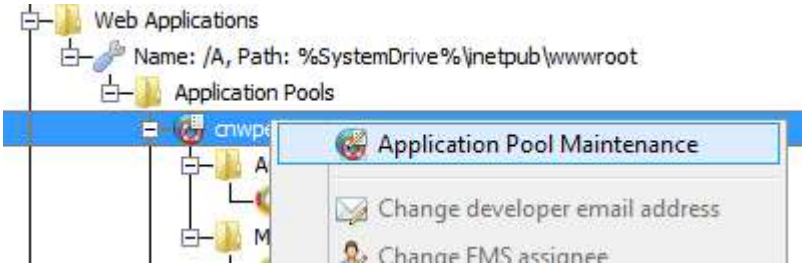


Declaring an explicit value will change the text colour from blue to black and the new value is held on the website object. Setting this value to 0 will remove the value from the website object and default back to the global setting.

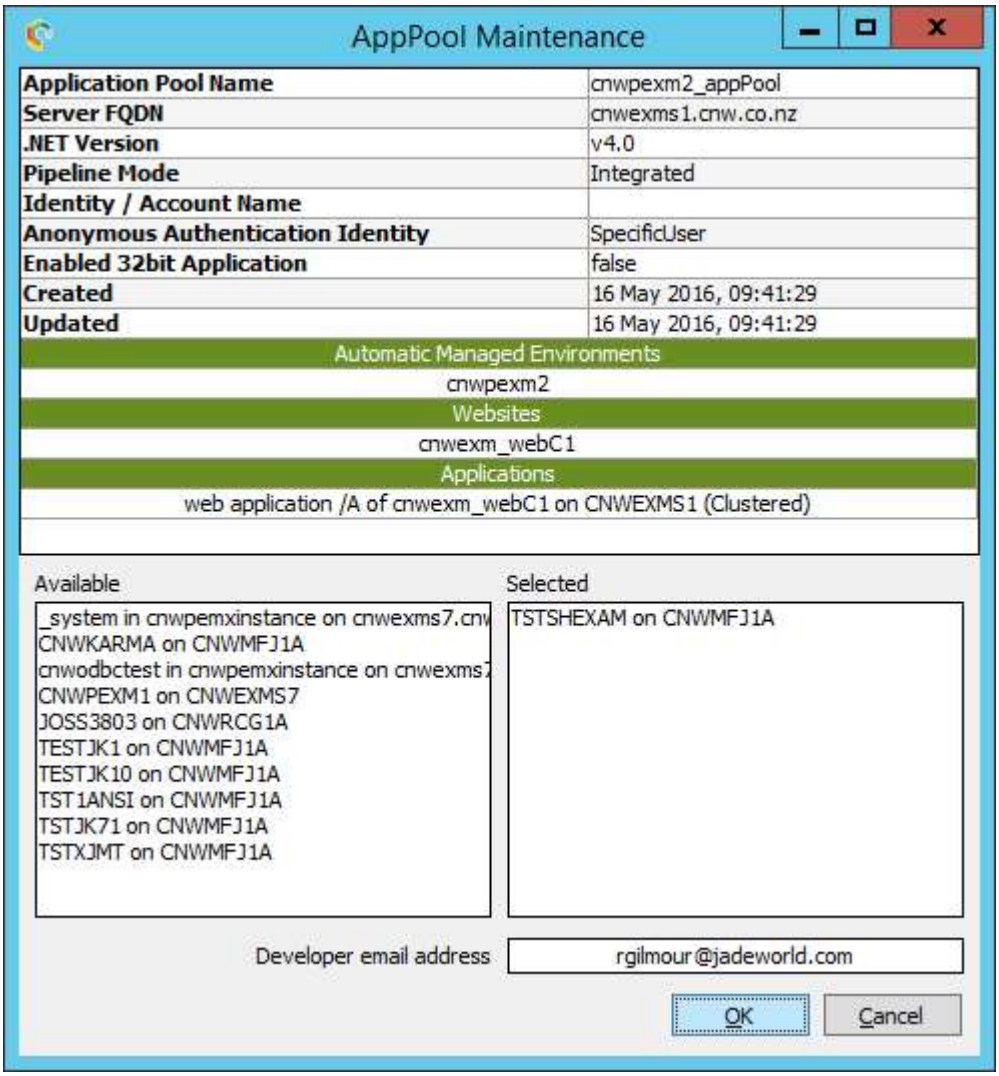


In the image above you will note two categories for Managed Environments. Environments listed under ‘Automatic’ have been automatically discovered during the discovery process. We have however provided a facility to ‘Manually’ associate a Managed Environment to a website.

To do this, right click on the AppPool. In the case above it is ‘cnwpexm2_appPool on CNWEXMS1’ The popup menu will be displayed and now ‘Application Pool Maintenance’ will be available



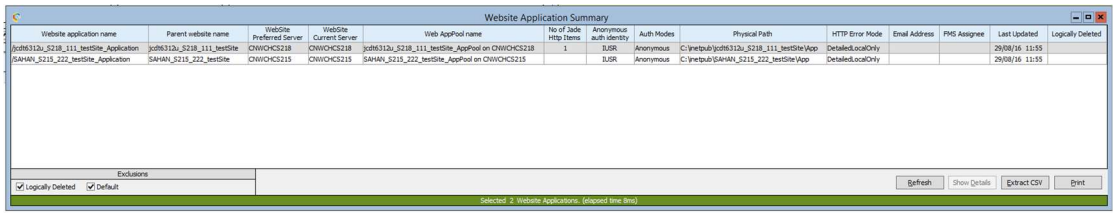
Clicking this will display the following form where you will get some further information on the AppPool but also have the ability to manually associate Managed Environments with the AppPool and change the developer email address for the selected AppPool.



You can also get to this form directly by double clicking the desired AppPool in the **AppPool Summary** form.

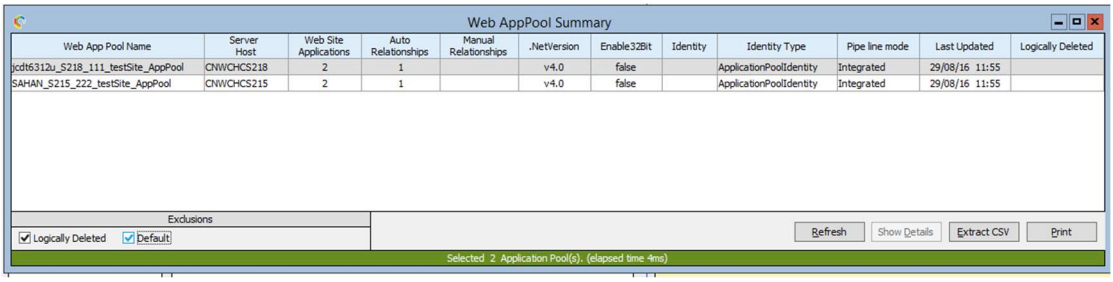
Double clicking certain entries (in all of the discovery forms using a tree display) will take you immediately to that objects summary. For example, when viewing **Website Summary** if you double click on a certificate it will open **Certificate Details** for that selection. Double clicking an AppPool in **Website Summary** will take you to the summary screen for that AppPool.

Web Site Application table view summary form will show discovered web site applications for web sites.

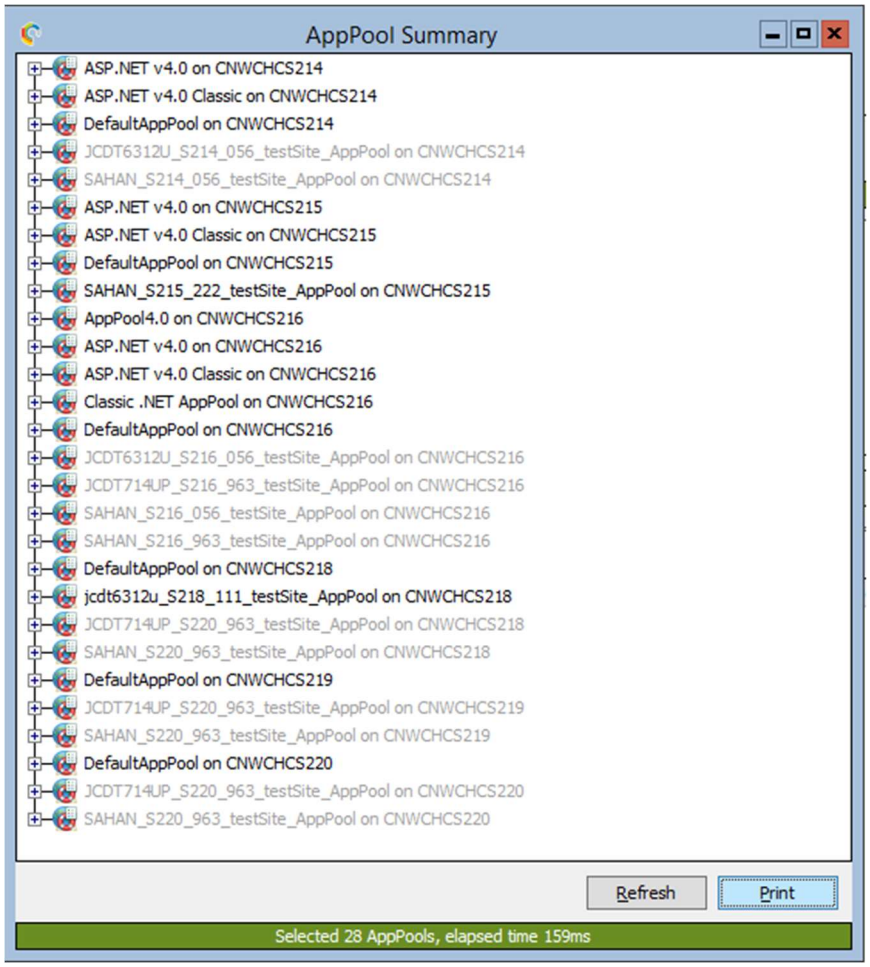


To get more information about selected item use ‘Show Details’ button or double click on it. Which will show its web site information.

Web AppPools table view **s**ummary

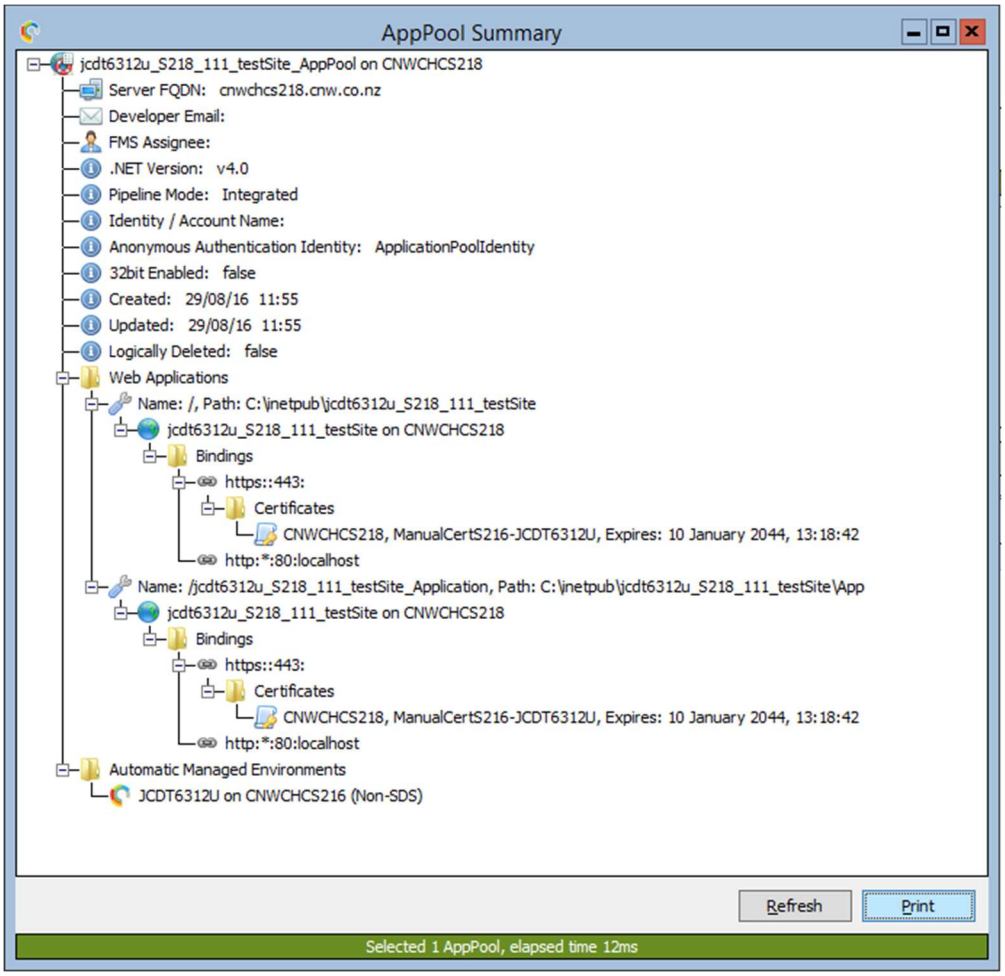


Web AppPools tree view **s**ummary

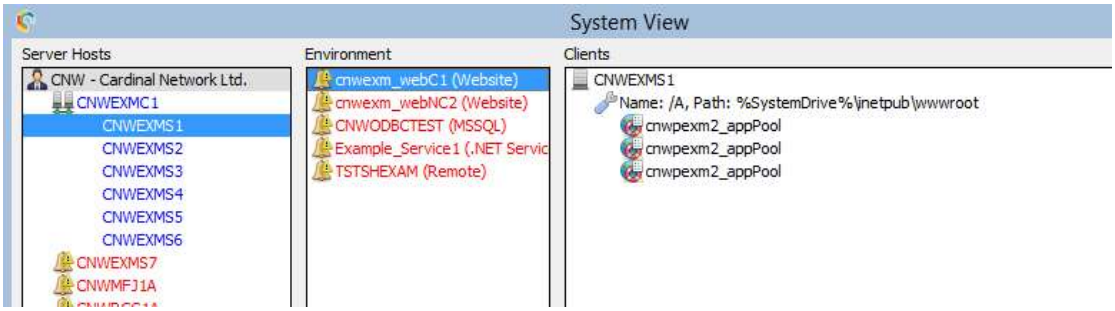


To get more information in,

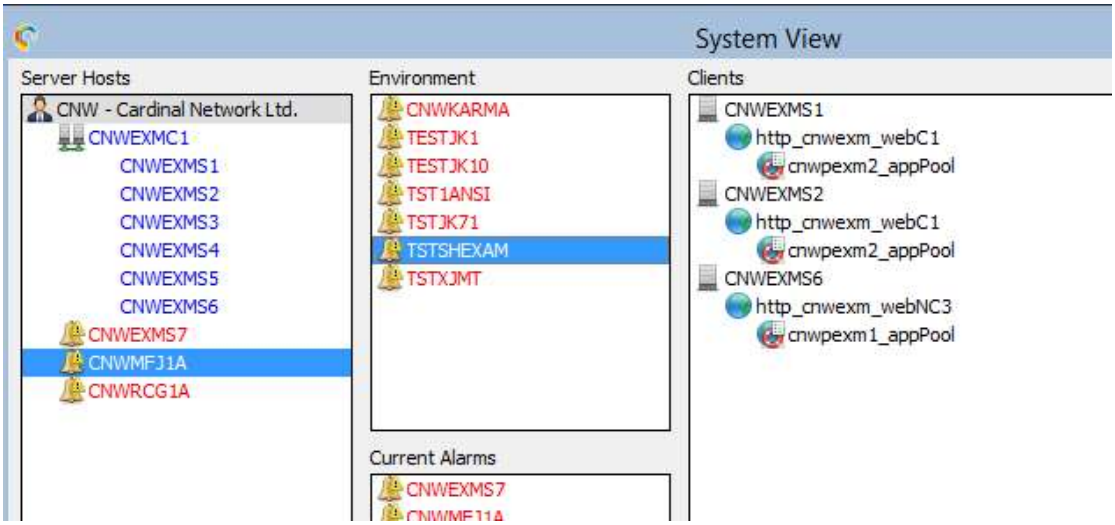
- In Web AppPools table view selected item use ‘Show Details’ button or double click on it. Which will show its web site information.
- In Web AppPools tree view selected item expand it.



System View now displays Remotes, Websites and .NET Services



Clicking on a Managed Environment that has an association with a website will display website and application pool information also



Double clicking on a website or application pool in the 'Clients' window on **System View** will bring up that objects summary form.

The following forms have had discovery information added and should provide a good overview of discovery entities in relation to

- Customer (**Customer Maint**)
- Server Host (**Server Host Maint**)
- Managed Environment (**Jade Environment Maint**)
- ODBC Monitored Database (**ODBC Maint**)

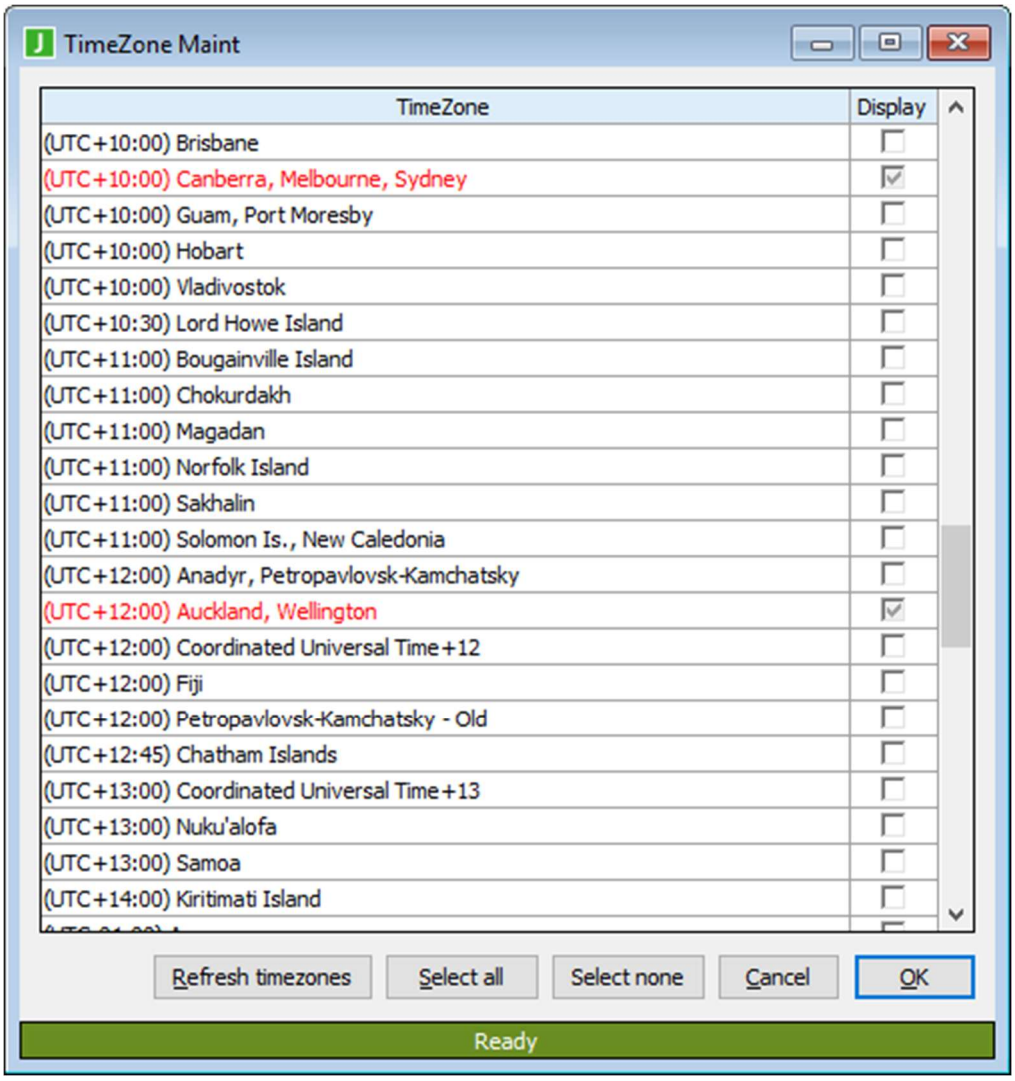
Timezone Maintenance

Selected by clicking the Setup→TimeZone Maintenance menu item.

This form allows you to globally select only the timezones you have an interest in, thereby reducing "noise" when populating various GUI elements in JSM with a timezone choice.

Timezones are derived automatically from the Windows OS registry and are periodically refreshed. You can force a refresh instantly by clicking the 'Refresh timezones' button.

Items shown in red are "in-use" items and cannot be unselected.



Webhook Endpoint Processing

Introduced in JSM 4.1.02, this feature allows JSM to act as a webhook client with no fixed processing structure. Webhook payloads, when received, are authenticated with a token, and subsequently matched to a user-defined script which knows how the payload should be interpreted.

This feature has a dependency on CnwSkyBound.dll v1.0.6 or later which must exist in the JSM server-side c_bin folder.

Token Maintenance

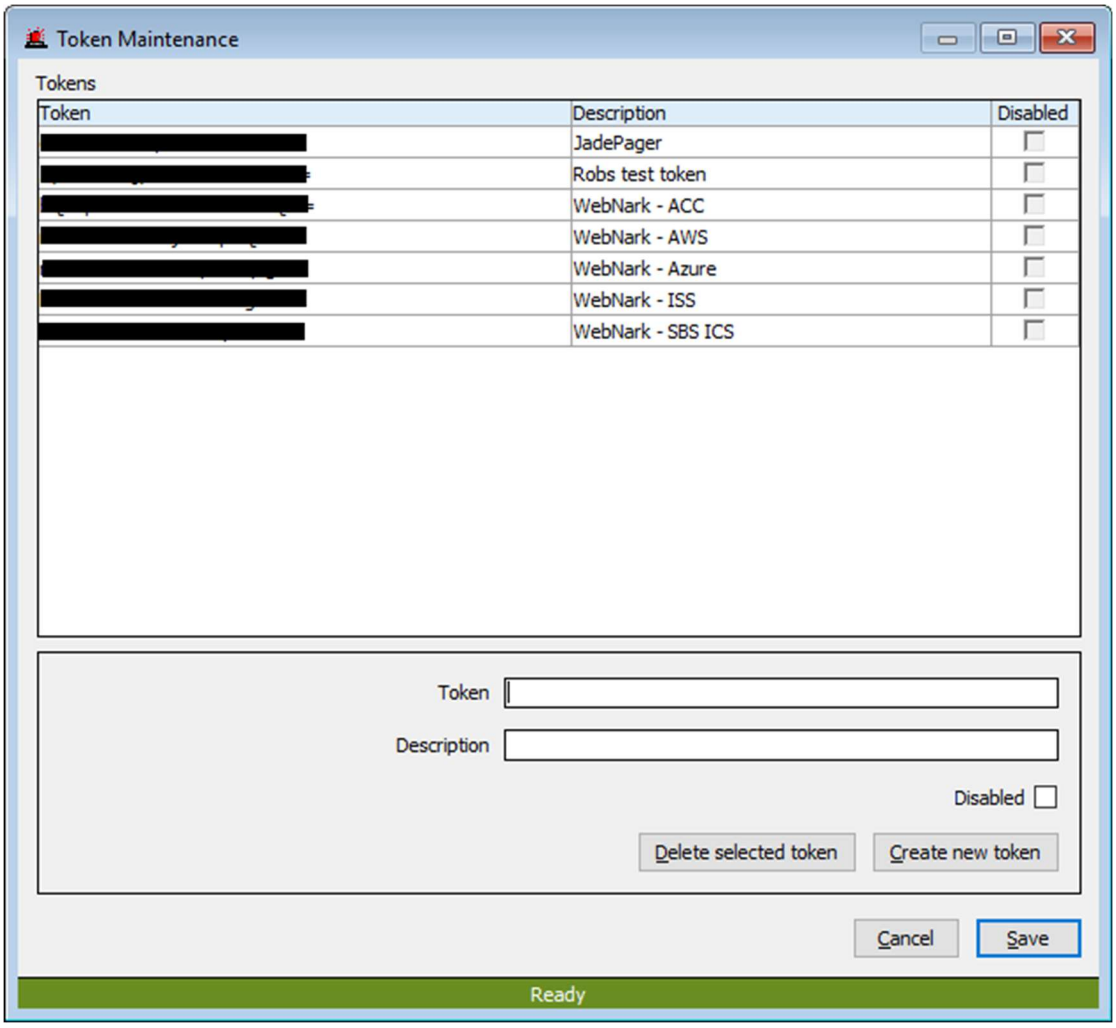
Tokens to be used by the Webhook Endpoint feature need to be created by JSM and issued to the administrator setting up the cloud service webhook. This is very important because JSM will not authenticate tokens it doesn't know about.

The token is to be provided as part of the Webhook URL, for example:

https://jcsm.jadeworld.com/bin_public/jadehttp.dll/WebNarc20/token=XyUaJmsLQyTHBFyzXWdgCs==/?JossCloudServiceManager

Tokens are generated by selecting the menu item Setup→Cloud Services→Token Maintenance.

All token maintenance changes are audited.



Create new token

Creates a new token. Once you've given the token a description, click the 'Save' option to commit the token to the database. This token can now be issued to the webhook administrator and JSM will accept payloads from that webhook.

Tokens are immutable. They can not be changed once they have been created. This is because external webhooks maybe using a token and deleting the token will break the webhook.

Token descriptions and the enabled/disabled state of a token can still be changed after creation.

When a token is created or, an existing token is selected, the complete URL for the webhook is automatically placed in the clipboard. This URL can then be sent to the webhook administrator as a simple paste operation.

Delete selected token

Deletes the selected token. Confirmation is required. If you plan on deleting a token, you should always contact the webhook administrator and let them know!

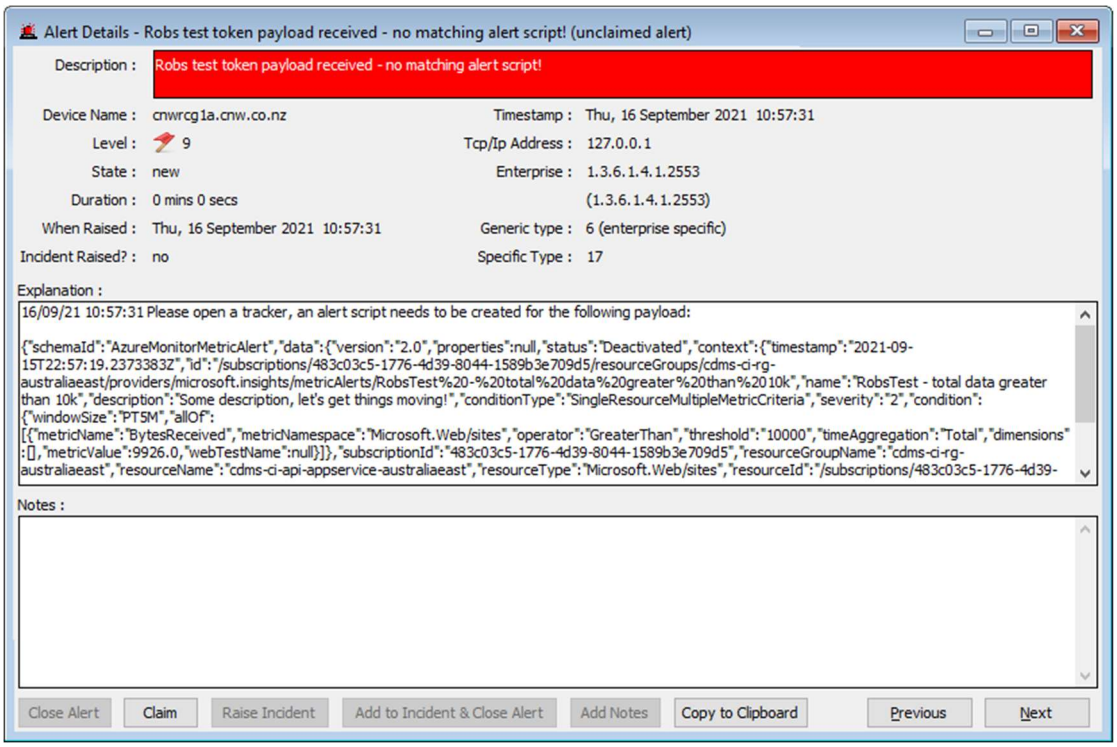
Disabled

Disables the selected token. If a webhook payload is received with a disabled token, then no processing of that payload occurs.

Script Builder

Webhook payloads are of an unknown structure, they could come from any number of cloud services, each with a unique payload. To facilitate the processing of these payloads JSM now contains a script builder.

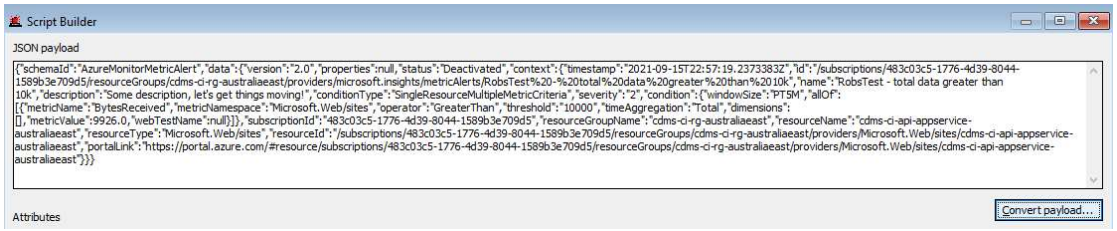
When a webhook payload arrives JSM attempts to find the JADE transient script associated with that specific payload. When a script cannot be found, JSM will raise an alert:



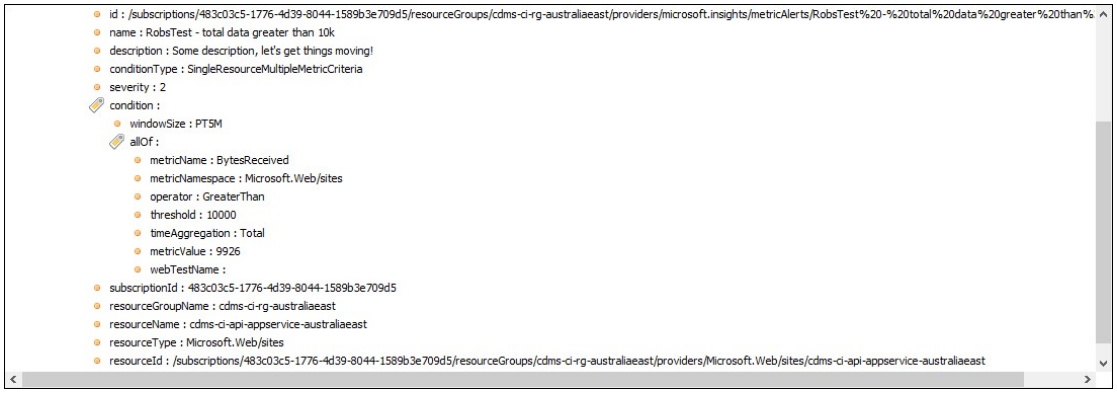
The alert contains the description associated with the token received and, the webhook payload received in JSON format. The user needs to copy the entire JSON string from the starting '{' to the ending '}' into their clipboard. This payload is then pasted into the Script Builder.

To access the Script Builder select Setup→Cloud Services→Script Builder.

Paste the JSON from your clipboard into the **JSON payload** section of the Script Builder form and click **Convert payload…**



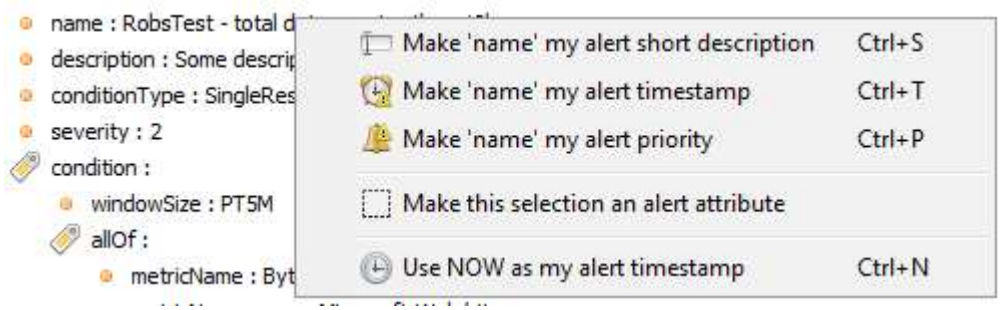
If the JSON is valid then the JSON is extrapolated into fields:



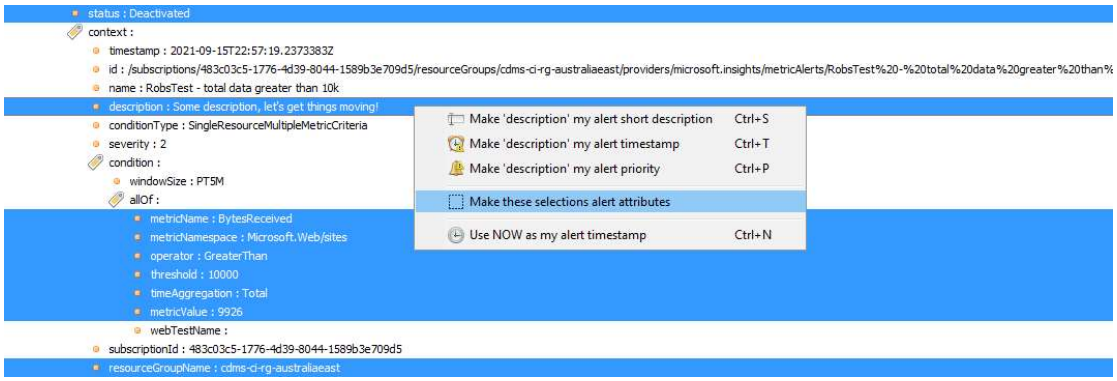
Any of these fields can now be selected to build the JADE transient script specific to this webhook payload. To do this use the right-click popup menu items.

In this example I have right-clicked over the 'name :' field and I will make 'name' my short description.

I will then repeat this over 'severity :' and make that field my alert priority and so on.



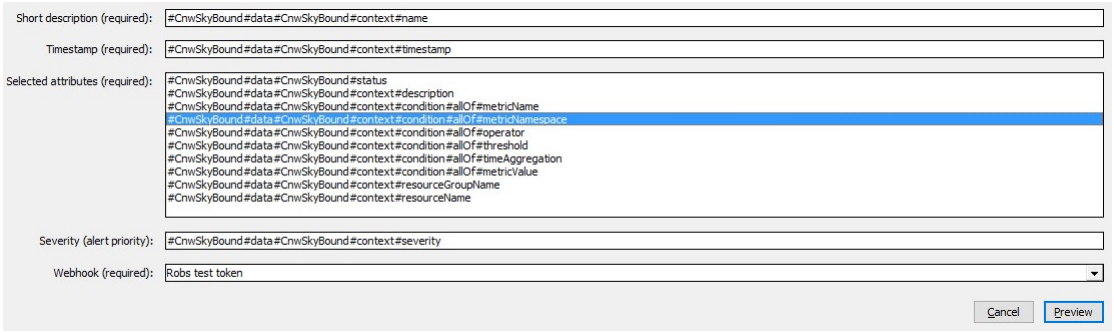
Any fields can be used to make up the alert body. You can multi-select by dragging your mouse across attributes or individually select them using CTRL+Click.



Once you've made some selections, click the **Make these selections alert attributes** menu item to add to the script.

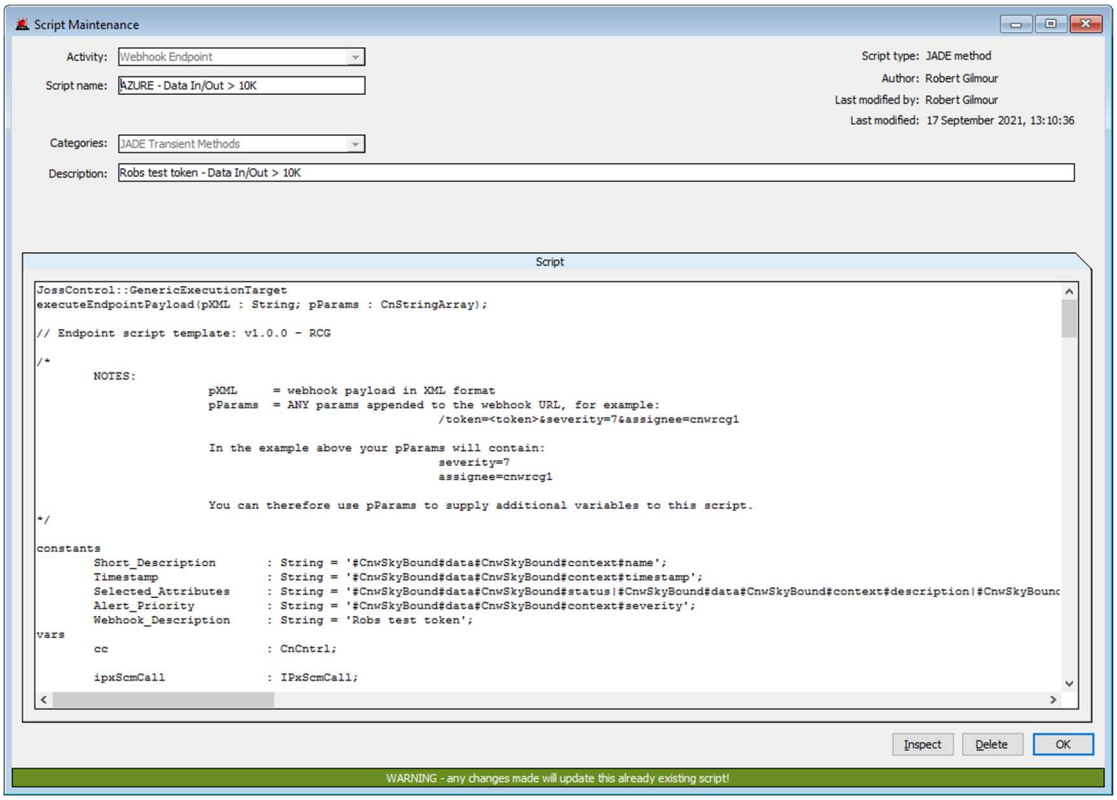
You can repeat this process as much as required to build up attributes for the alert body.

The ordering of attributes can be changed in the script builder using drag and drop. To do this select the attribute you want to move with the mouse and, while holding the left-mouse button down, drag the item up or down.



In the example above we have provided everything we need create our JADE transient script. Be aware that nothing is committed yet!

Click the **Preview** button to open the **Script Maintenance** form.



All the information you've supplied in the **Script Builder** has now been applied to the JADE script template.

By default, alerting is enabled in the script template. There are also commented out sections for creating a Tracker, Email and Paging. As this is a JADE script everything can be customised, including additional alert text, alert priorities, mail, or paging recipients etc.

The script can be further tailored to accept specific webhook supplied variables. These are passed into the script via the pParams variable in the method signature. The best way to explain this is to look at the webhook URL:

https://jcsm.jadeworld.com/bin_public/jadehttp.dll/WebNarc20/token=XyUaJmsLQyTHBFyzXWdgCs==&severity=9&assignee=cnwrcg1&paging=true/?JossCloudServiceManager

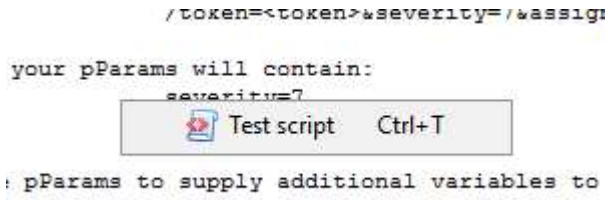
Notice that I've supplied some additional parameters to the URL separated by the & character. These additional parameters are passed into the JADE script when it's executed however it is up to you to code for them.

Additional parameters above must be defined AFTER the token. The token should always be first otherwise the payload will be unauthenticated and ignored.

The script will work as-is for alerting without requiring any JADE programming knowledge. DO NOT mess with the script if you don't know what you're doing.

Before you can commit the script, you must supply a script name, description isn't a requirement, but it is recommended as you could potentially have many scripts to process webhook payloads.

When working on your script, you can test it on-the-fly. The script doesn't have to be committed for it to be tested. To test the script, right-click over the script body and select **Test script**:



JSM will let you know if the script was executed successfully but it is up to you to determine the results of your script.

To commit the script, click the **OK** button. Now whenever this webhook payload arrives your JADE script will be executed to process it.

Some things to note:

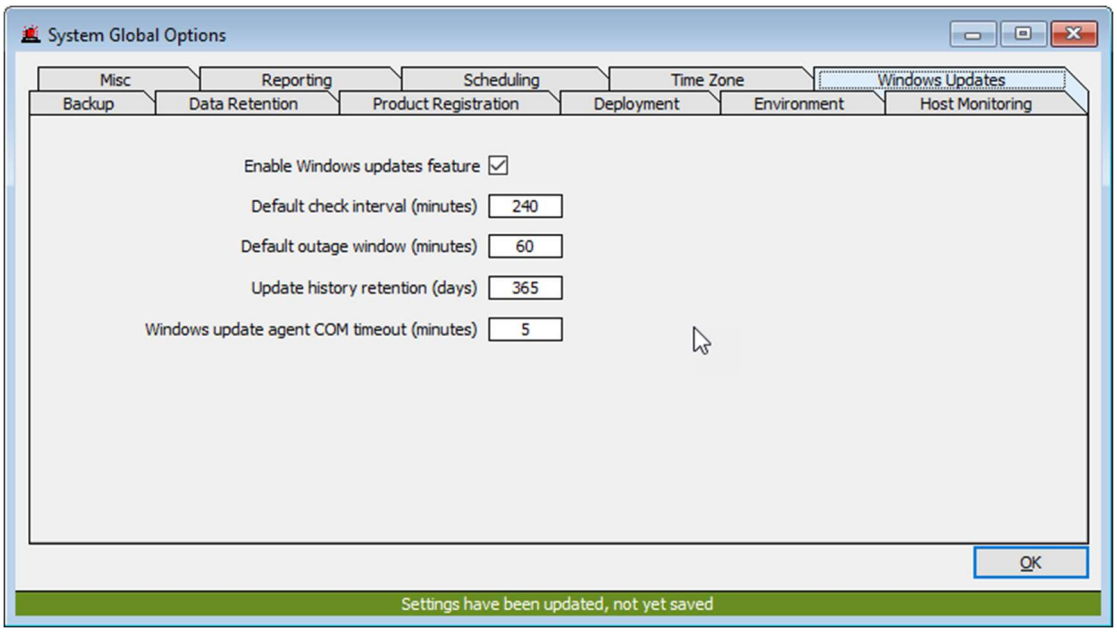
- Webhook Endpoint scripts can never be created without going through the **Script Builder**.
- Existing scripts can be modified by using the Setup→Scripts menu item, selecting Webhook Endpoint as an activity and selecting your script for editing.
- Or alternatively, you can find your scripts by selecting Setup→Script Categories and looking under the JADE Transient Methods section.

MS Windows Automatic Updates (Patching)

Introduced in 4.1.0.2 JSM will support the scheduling and installation of MS patches. JSM will also catalogue MS patch information and provides new forms to summerize this data.

The Windows Update feature is only available to JADECare24 and is disabled globally by default. To enable the Windows Update feature, navigate to System Global Options.

System Global Options



Enable Windows updates feature

Makes JSM capable of scheduling and managing MS patch installations. It does not turn on the feature for each ServerHost.

Turning off this feature will cause ALL ServerHosts with the feature enabled to have the feature turned off. All running instances of KarmaWindowsUpdateManager will be instructed to terminate.

The following values will be populated into the ServerHostMaint Windows Updates area when setting up Windows Updates for a specific ServerHost the first time.

Default check interval (minutes)

This value determines how frequently checking for updates will be performed.

Default outage window (minutes)

This value determines the allocated time in which all patch installations and reboots must occur.

Update history retention (minutes)

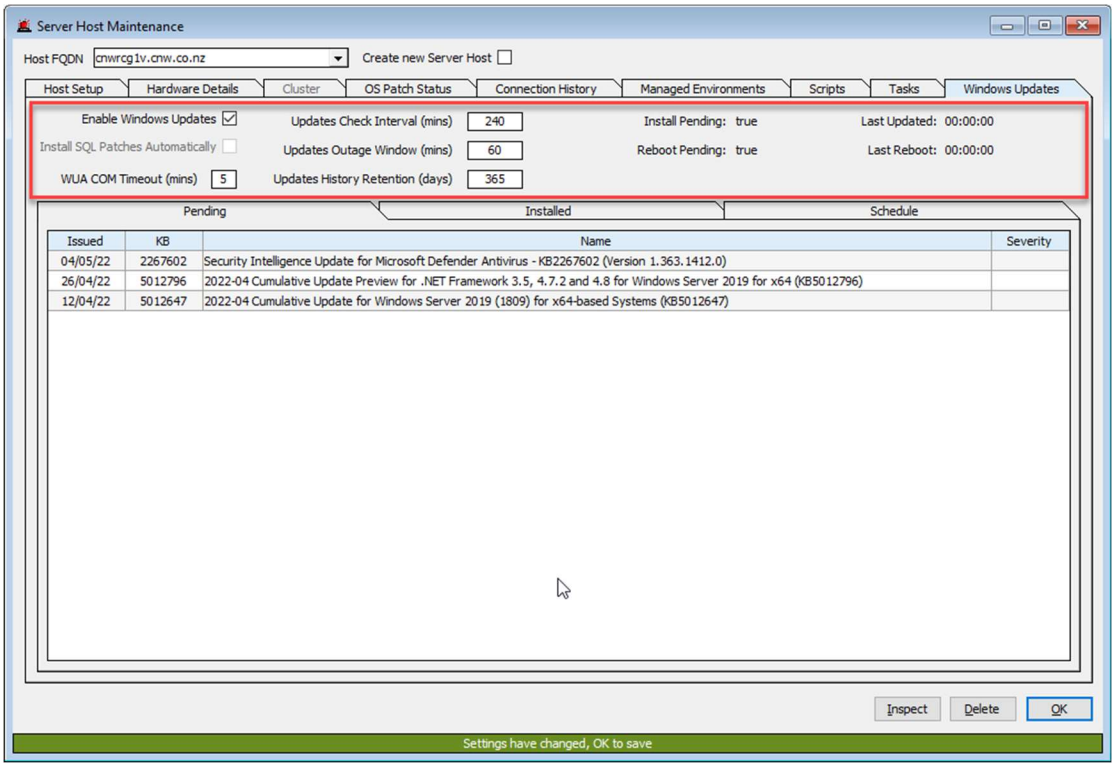
This value determines how long activity log and patch information will be retained.

Windows update agent COM timeout (minutes)

This value determines how long we should wait on the WUA to respond before giving up. The Windows COM interface can be a funny beast so this timeout capability is essential.

Server Host Maintenance

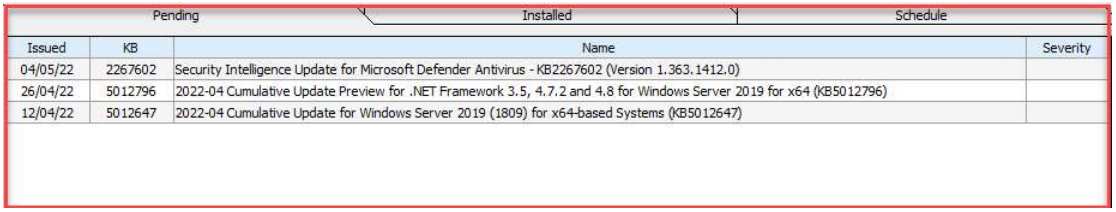
The Windows Update feature, settings and installation schedule are uniquely associated with each Server Host. To use this feature, you need to set it up in Server Host Maintenance.



As you can see from the above image, the System Global Options are replicated here and this ServerHost is using those default values. Once saved these values are associated only with this instance of ServerHost. Changes are audited.

Pending

Updates pending for this ServerHost are shown below. You can double click on an update for more information about the update. This is true for any of the new forms where updates are represented in a table like below.



Installed

Historical patch installation data for this ServerHost.

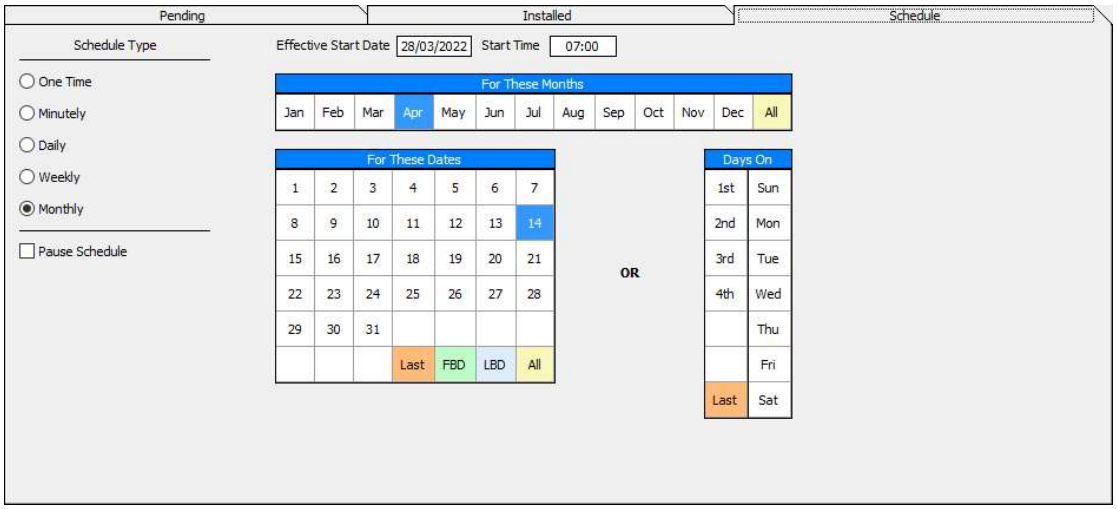
Schedule

Extensive scheduling options are available to setup automatic Windows Updates. These range from a one-off through to a recurring schedule for a particular date (or dates) in a month (or months) year upon year.

NOTE: if the Windows Update manager is already in a patch cycle when the schedule fires the schedule run will be ignored.

Provision has been made to pause the schedule at any time whilst still retaining the schedule setup. Similary, if the Windows Updates feature is turned off the schedule setup will be preserved.

In 4.1.0.2 we've introduced a redesigned scheduling interface. Multi-selection of months, dates and days is supported by holding down the CTRL key while clicking selections in each table.



Schedule Type – One Time
A one-off schedule that will run once and never repeat again.

Schedule Type – Minutely
Recurs every minute on a value defined by the user.

Schedule Type – Daily
Occurs every day at a given time.

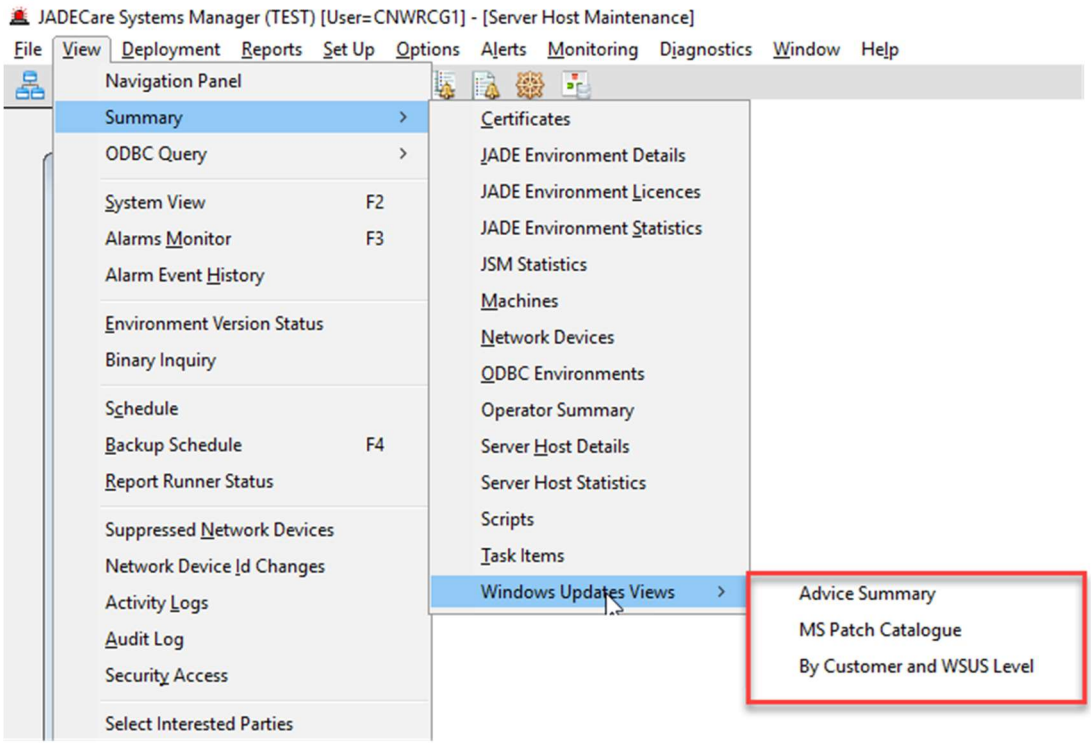
Schedule Type – Weekly
Recurs every week on days defined by the user.

Schedule Type – Monthly
The most complete and granular scheduling option that recurs for the months requested on any user-defined dates or days.

- LAST – last day of the month.
- FBD – first business day of the month.
- LBD – last business day of the month.

Summary Forms

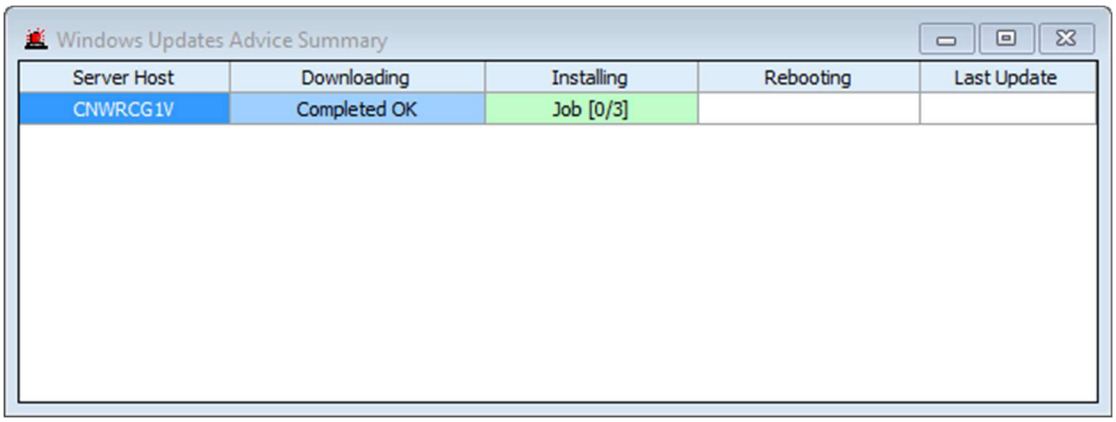
Some summary forms have been added to JSM to give an overview of Windows updates. These are accessible here:



Advice Summary

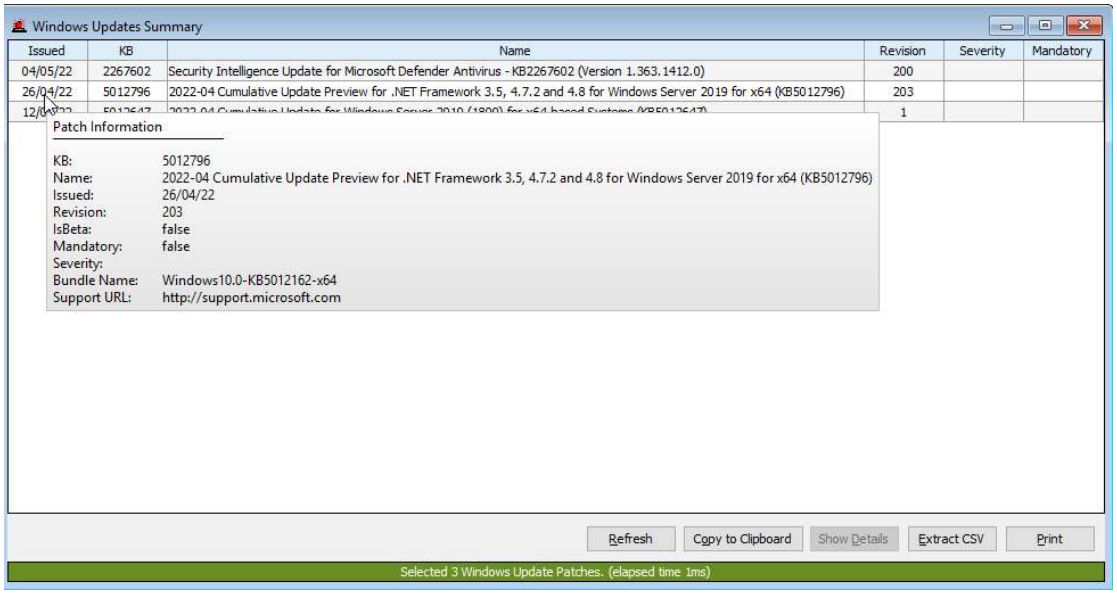
Provides a very basic graphical overview of Windows Updates activity, in a similar vane to the Deployment Summary form. ServerHosts will be added and removed dynamically as MS patch information is received during downloads, installations, and reboots.

The form has primarily been provided so activity can be monitored by a 24/7 1st level support.



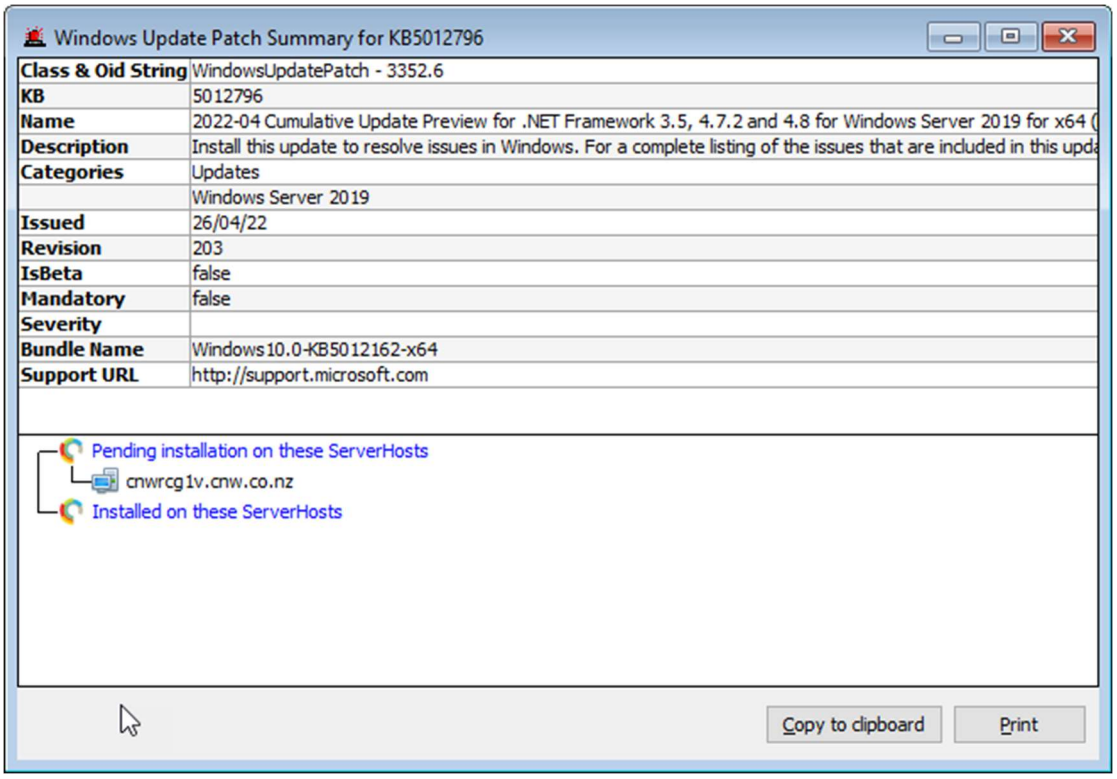
MS Patch Catalogue

Every MS patch that JSM knows about, independent of ServerHost. Bubble help is available for each patch and double clicking on patch information is supported.



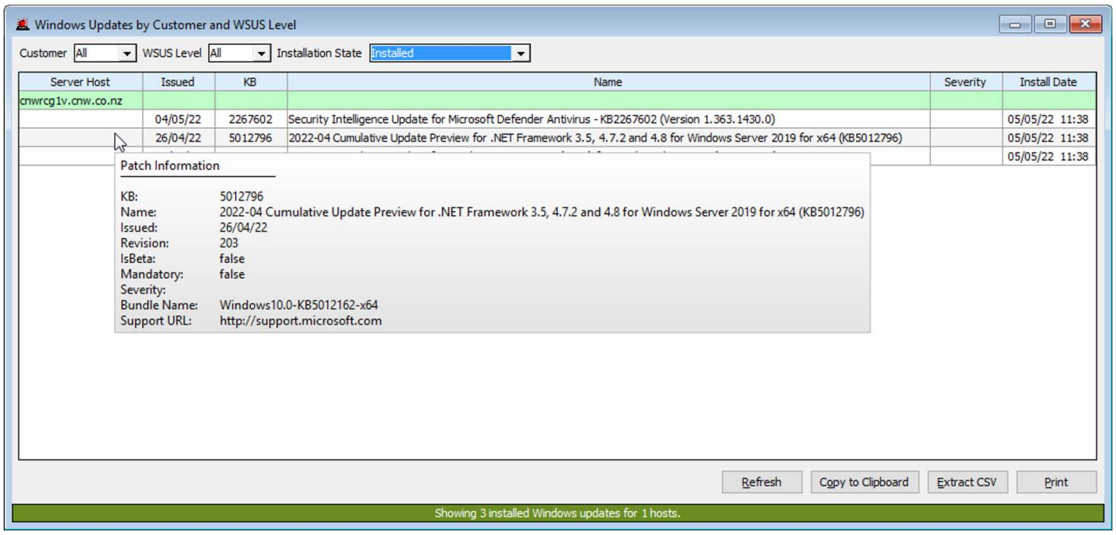
Double-clicking a patch presents a sub-view form. This form supports hyperlink clicking to the support webpage for the selected MS patch. Double-clicking on the ServerHost entries in the bottom control will take you directly to the selected ServerHost maintenance form.

This sub-view from is accessible from anywhere patch information is displayed in a table.



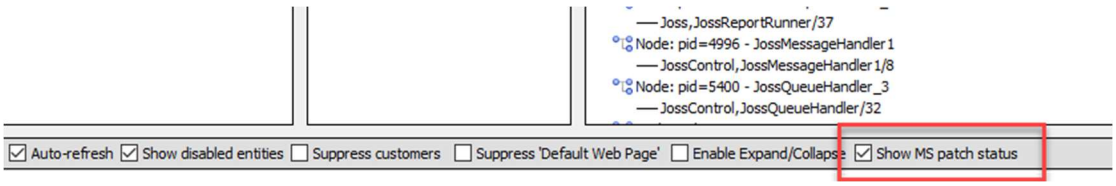
By Customer and WSUS Level

Provides the ability to filter patch information applicable to both customer, WSUS level and installation state. Bubble help is available for each patch and double clicking on patch information is supported.



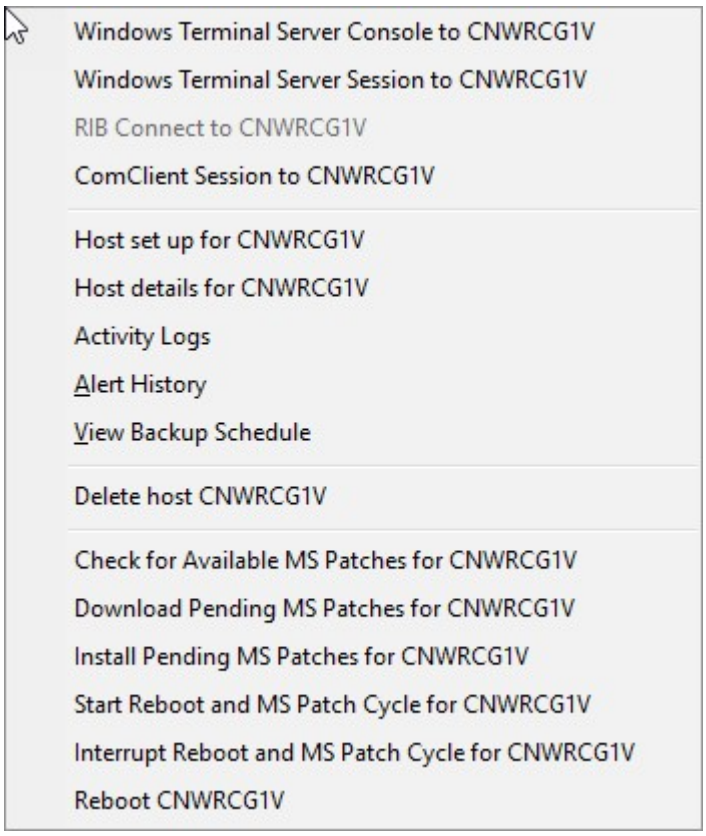
System View

System View includes a new checkbox item. When this item is checked ServerHosts that have Windows Updates actions pending are highlighted with a light green background.



New popup menu items are available when selecting a ServerHost. This enables the user to execute Windows Update options adhoc and are independent of any scheduled activity.

NOTE: if Windows Updates operations are already in play user requests may be ignored (these scenarios will be logged).



Check for Available MS Patches
Initiate a check to see if any new patches are available for this ServerHost. Activity is logged.

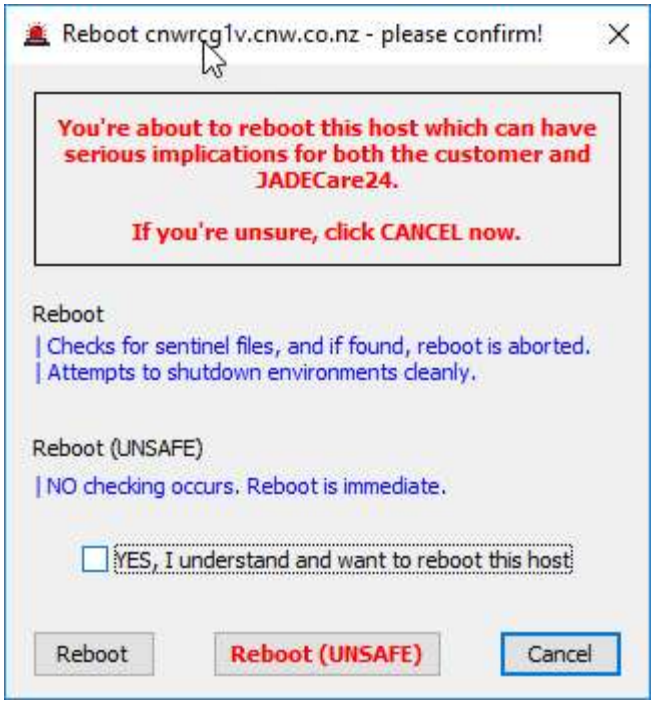
Download Pending MS Patches
Initiates a download of any available patches. Limited usefulness when dealing with managed ServerHosts under WSUS control. Activity is logged.

Install Pending MS Patches
Initiates the installation of patches. Does not include reboots that may be required because of patch installation – keep this in mind! Activity is logged.

Start Reboot and MS Patch Cycle
An adhoc patch and reboot cycle, the same behaviour as a scheduled run. Reboots will occur automatically if required. Confirmation is required. Activity is logged.

Interrupt Reboot and MS Patch Cycle
Notifies the current operation to halt. Only applicable during download and installation activities. If there are 10 patches to be installed the process will interrupt after the current patch has finished installing. Activity is logged.

Reboot
Reboots the ServerHost. Confirmation is required. There is a SAFE option and an UNSAFE option. Activity is logged.



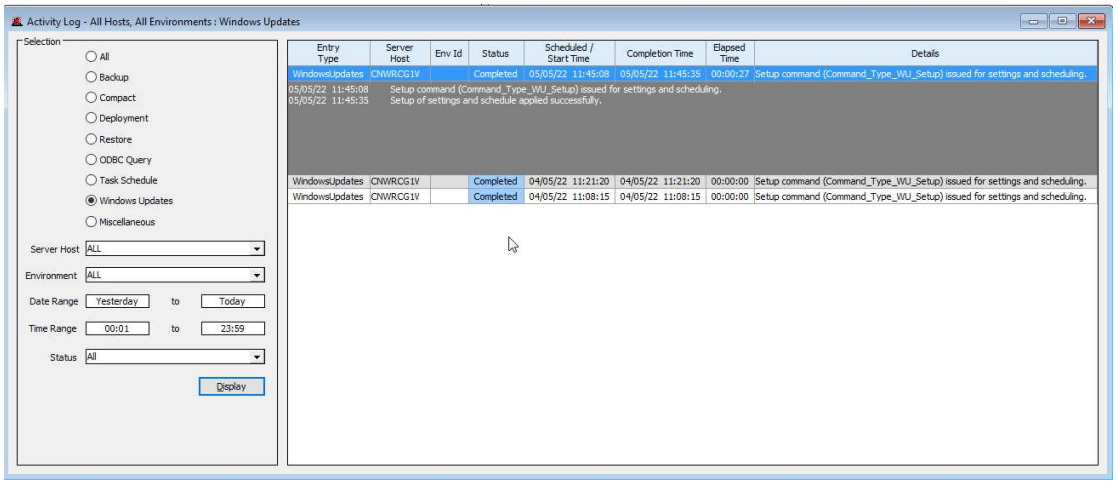
Reboot
This is the safe option. Sentinel files are considered, and environments are shutdown cleanly before a reboot can occur. Present sentinel files or being unable to shutdown an environment will prevent the reboot from occurring.

Always choose this option if possible.

Reboot (UNSAFE)
Instant reboot. Nothing will be checked, anything running will be terminated immediately.

Activity Logging

The activity logging facility now includes Windows Update activities.

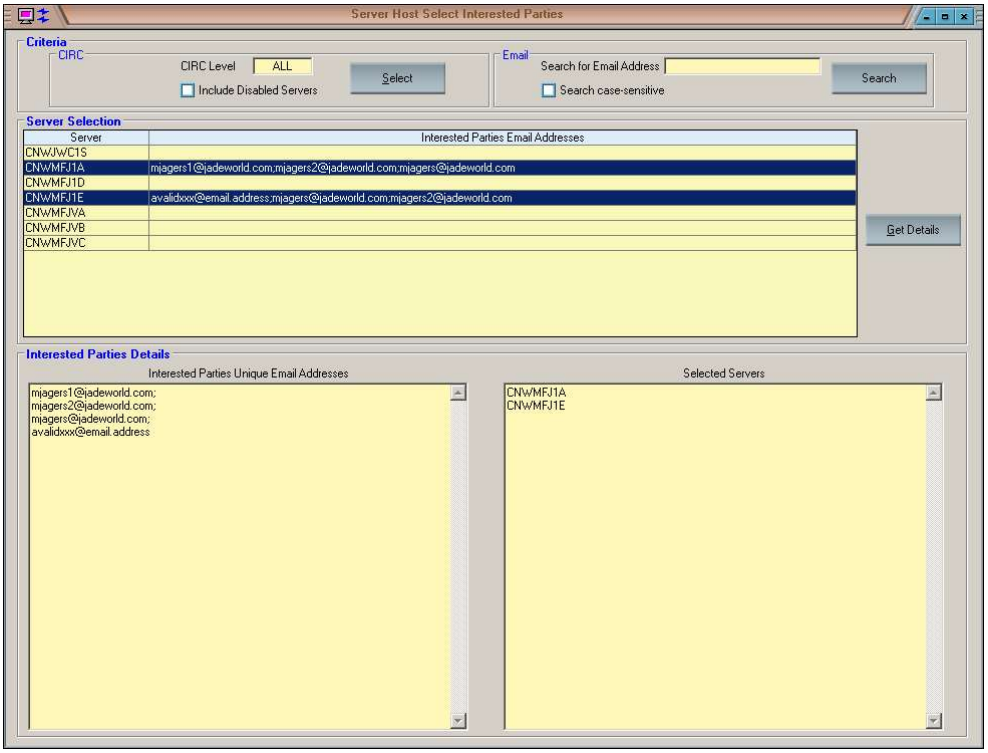


Select Interested Parties

The **Server Host Maintenance** form can be used to define the CIRC Level and also a list of email addresses for parties with an interest in this server.

The form **Select Interested Parties** can be used to:

- identify all servers and unique email addresses of interested parties
- locate which servers have defined the email address (partial / full) within the interested parties list



Two different criteria can be used to select the servers concerned:

1. A specific CIRC Level or ALL and may include or exclude servers marked as disabled.
2. A partial or full email address

Either of the above two criteria will build a table of servers that match the criteria. Double-clicking on a row of this table will cause the **Server Host Maintenance** form to be displayed for the associated server. Changes made to the'Interested Parties' and committed by the maintenance form will cause the relevant entries to be modified.

One or more rows may be selected. If no rows are selected, then the default will be that all rows are implicitly selected.

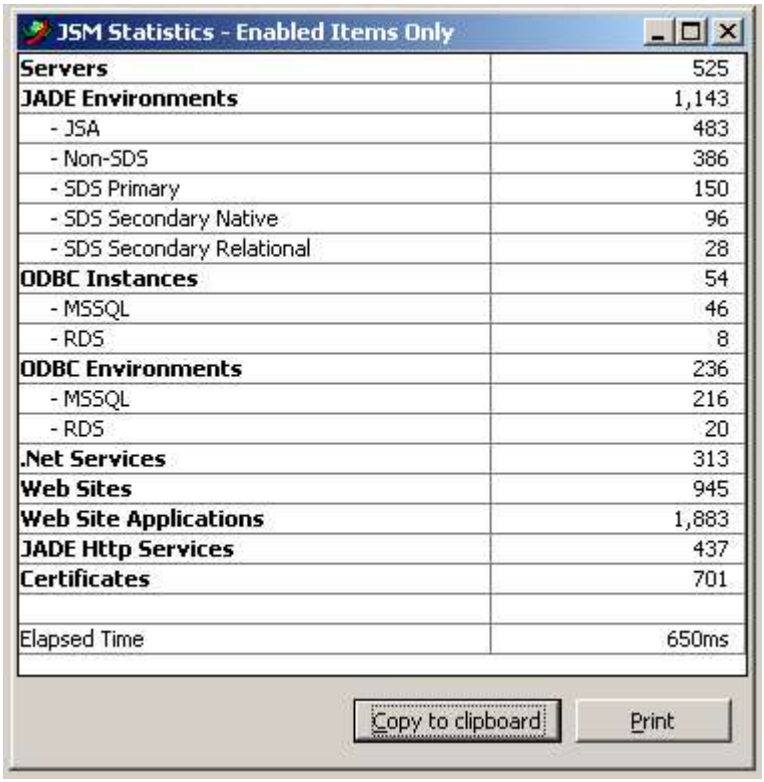
Clicking on the **Get Details** button will update :

- Interested Parties Unique Email Addresses text box with unique email addresses for the servers 'selected'
- Selected Servers text box with a list of the server names selected

Setting focus on either of the two text boxes with details will cause all the text to be selected and this can be copied and pasted as required.

JSM Statistics

This form is displayed via menu options View | Summary | JSM Statistics. It displays the number of various core entities managed by this JSM instance.

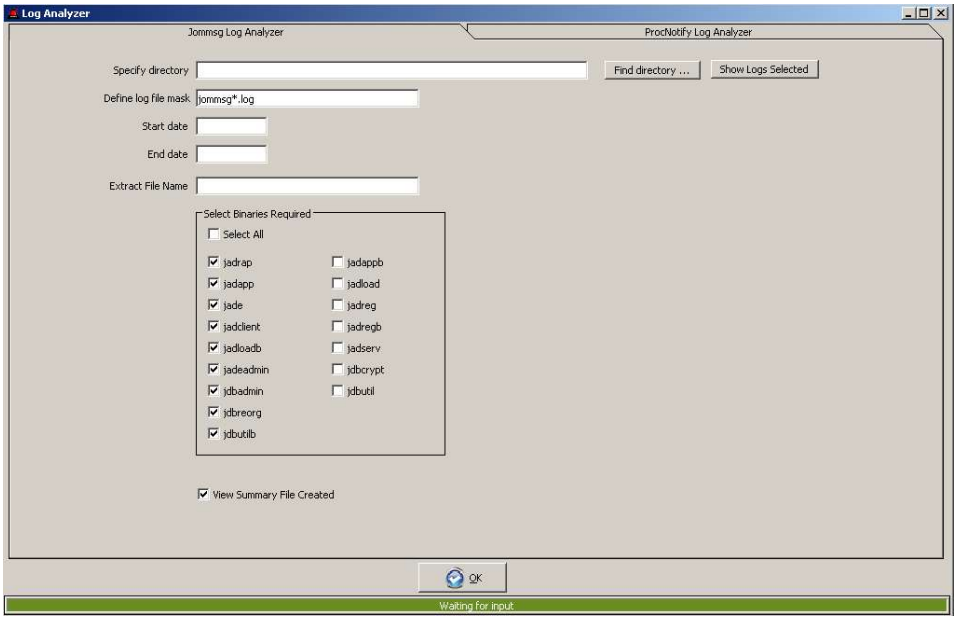


Diagnostics – Log Analysis

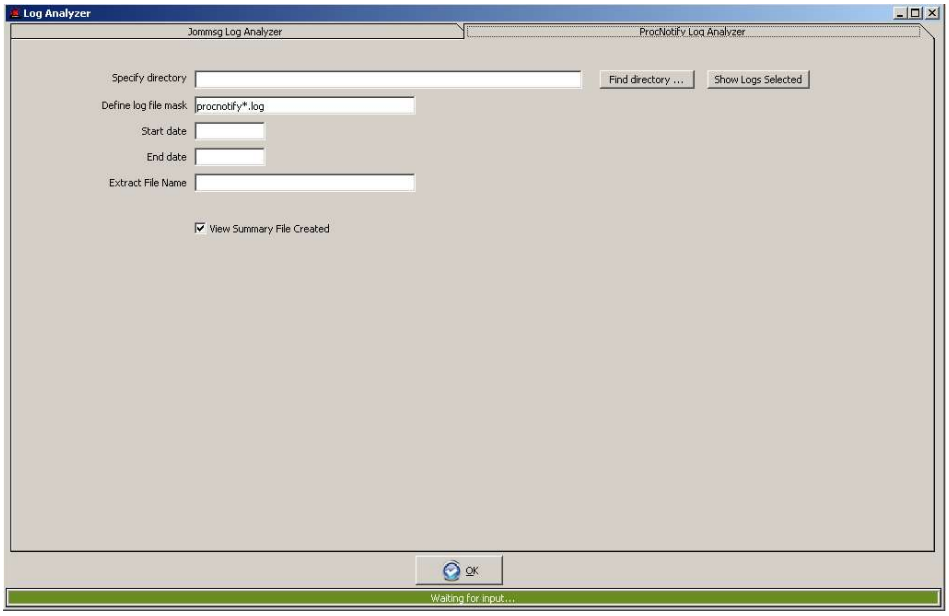
The menu item **Diagnostics** has a sub-menu item **Log Analysis**. This provides a form with two sheets that can be used for:

- Parsing one or more jommsg.logs producing a summary CSV formatted file containing the start and end details for specific JADE executables.
- Parsing one or more procnotify.logs producing a summary CSV formatted file detailing:
    - PID
    - Start and end times of the executables
    - The CPU utilization (if end task logged)
    - The exitcode (if end task logged)
    - Account used
    - Command line

Jommsg Log Analysis



ProcNotify Log Analysis



Appendix 1 – JAWS Control File

The following is an example of the JADECare Systems Manager JAWS control file as of the 3.9.02 release. Please note that this example contains the Tracker background applications.

```
[JawsOptions]
JawsVersion=8, 9, 0, 11
FileName=c:\josstest\logs\jaws.log
AppendJawsIdEndJade=true

[JadeServer]
ServerName=josstest
Path=d:\josstest\server\c_system
Exe=d:\josstest\server\c_bin\jadrap.exe
Ini=d:\josstest\server\c_bin\josstest.ini
Launch=True
Priority=Normal

[JossMessenger]
Schema=JossControl
App=JossMessenger
Exe=e:\josstest\server\c_bin\jade.exe
Path=e:\josstest\server\c_system
Ini=e:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=True
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossMessageHandler1]
Schema= JossControl
App=JossMessageHandler1
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossGeneralEventMgr]
Schema= JossControl
App=JossGeneralEventMgr
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossSnmpMgr]
Schema= JossControl
App=JossSnmpMgr
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
```

```
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossHeartBeat]
Schema=JossControl
App=JossHeartBeat
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossFileTransferMgr]
Schema=JossControl
App=JossFileTransferMgr
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossQueueHandler_1]
Schema=JossControl
App=JossQueueHandler
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossQueueHandler_2]
Schema=JossControl
App=JossQueueHandler
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossQueueHandler_3]
Schema=JossControl
App=JossQueueHandler
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI
```

```
[JossResponseMonitor]
Schema=JossControl
App=JossResponseMonitor
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossMailMgr]
Schema=JossControl
App=JossMailMgr
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossStatsCollector
Schema=JossControl
App=JossStatsCollector
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossDiscovery]
Schema=JossControl
App=JossDiscovery
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossReportRunner_1]
Schema=Joss
App=JossReportRunner
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JossReportRunner_2]
Schema=Joss
App=JossReportRunner
Exe=D:\josstest\server\c_bin\jade.exe
```

```
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[ScmAdvisor]
Schema=ScmControl
App=ScmAdvisor
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[ScmEscalationHandler]
Schema=ScmControl
App=ScmEscalationHandler
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[ScmReminderHandler]
Schema=ScmControl
App=ScmReminderHandler
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[ScmBackground]
Schema=ScmControl
App=ScmBackground
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[ScmKeywordBuilder]
Schema=ScmControl
App=ScmKeywordBuilder
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
```

```
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[ScmWebServiceV1]
Schema=ScmSchema
App=ScmWebServiceV1
Exe=D:\josstest\server\c_bin\jade.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
Server=RemoteServer
Launch=true
ClientType=NormalClient
Priority=Normal
Title1=NOGUI

[JadeAppServer_11111]
Server=remoteServer
Exe=D:\josstest\server\c_bin\jadapp.exe
Path=d:\josstest\server\c_system
Ini=d:\josstest\server\c_bin\josstest.ini
AppServerPort=11111
name=AppServer
Launch=True
ClientType=ApplicationServer
Title1=Application Server - Tcp Port 11111
```