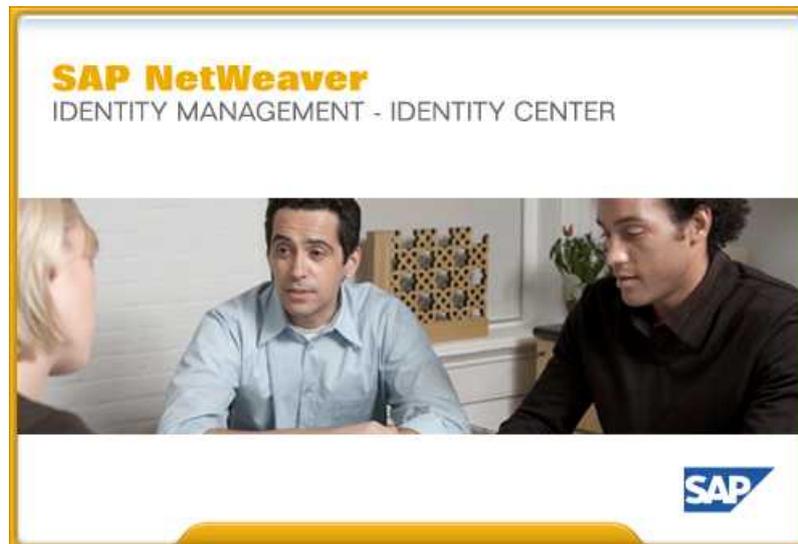


SAP NetWeaver® Identity Management Identity Center

Implementation guide

- Optimizing dispatcher performance



© 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, PowerPoint, Silverlight, and Visual Studio are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, z10, z/VM, z/OS, OS/390, zEnterprise, PowerVM, Power Architecture, Power Systems, POWER7, POWER6+, POWER6, POWER, PowerHA, pureScale, PowerPC, BladeCenter, System Storage, Storwize, XIV, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, AIX, Intelligent Miner, WebSphere, Tivoli, Informix, and Smarter Planet are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and its affiliates.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems Inc.

HTML, XML, XHTML, and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Apple, App Store, iBooks, iPad, iPhone, iPhoto, iPod, iTunes, Multi-Touch, Objective-C, Retina, Safari, Siri, and Xcode are trademarks or registered trademarks of Apple Inc.

IOS is a registered trademark of Cisco Systems Inc.

RIM, BlackBerry, BBM, BlackBerry Curve, BlackBerry Bold, BlackBerry Pearl, BlackBerry Torch, BlackBerry Storm, BlackBerry Storm2, BlackBerry PlayBook, and BlackBerry App World are trademarks or registered trademarks of Research in Motion Limited.

Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.

INTERMEC is a registered trademark of Intermec Technologies Corporation.

Wi-Fi is a registered trademark of Wi-Fi Alliance.

Bluetooth is a registered trademark of Bluetooth SIG Inc.

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Preface

The product

The SAP NetWeaver Identity Management is a general purpose identity management application which provides the functions and services needed to integrate distributed identity data in the system landscape to efficient, heterogeneous identity lifecycle management.

The reader

This manual is written for people who are to optimize dispatcher performance in a production environment.

Prerequisites

To get the most benefit from this manual, you should have the following knowledge:

- Thorough understanding of SAP NetWeaver Identity Management.
- SAP NetWeaver Identity Management 7.2 SP7 or newer.

The manual

This document describes how you configure dispatcher properties for optimal performance in a production system.

This tutorial is not a substitution for training.

Related documents

You can find useful information in the following documents:

- SAP NetWeaver Identity Management Identity Center tutorials and help file

Table of contents

Introduction	1
Dividing operations between dispatchers.....	3
Using the Windows runtime engine.....	4
Performance considerations	5
Controlling the number of runtime engines.....	5
Controlling job looping of runtime engines	6

Introduction

This document provides a description of how the dispatcher works, and some recommendations on how to configure the dispatchers in a production environment to achieve optimal performance.

Note:

The recommendations are based on a general usage scenario, and may not be applicable to all configurations. There is no guarantee what the effect of these recommendations is in a specific system. Each implementation must be assessed separately and the dispatchers configured accordingly.

The dispatcher has the following main functions:

- Executing regular jobs.
- Executing provisioning jobs.
- Processing tasks.
- Evaluating relations (assignments).
- Evaluating approvals.
- Executing the scheduled procedures for housekeeping.

Executing regular jobs

The dispatcher starts the runtime engine to check if there are jobs waiting to be executed. When there is a regular job to be processed, the runtime engine processes all entries for all passes in the job before completing.

Executing provisioning jobs

When running a provisioning job, the runtime engine processes one entry at a time and will continue processing as long as there are entries to process for this job.

Processing tasks

The dispatcher is responsible for performing the following operations:

- Expanding task groups (ordered/unordered groups)
- Evaluating conditional and switch tasks

Evaluating relations (assignments)

The dispatcher performs the following operations regarding relations (assignments):

- Triggering event tasks
- Handling validity for assignments

Evaluating approvals

The dispatcher is responsible for the following operations regarding approvals:

- Initializing approvals
- Sending notification messages
- Processing approve/decline events
- Handling escalations

Executing scheduled procedures for housekeeping

The dispatcher executes the scheduled procedures defined on the "Housekeeping" tab of the "Dispatchers" node.

Dividing operations between dispatchers

How many dispatchers you configure in the system depends on the load on the system and the distribution of the different operations that the dispatcher performs. The number and setup of the dispatchers also depends on requirements for high-availability and performance. The system landscape XL – Production described in the *SAP NetWeaver Identity Management Identity Center Installation overview* describes how to achieve high availability with duplication of the runtime servers.

High availability requires two dispatchers with identical configuration running on different servers.

The dispatchers could be configured like this:

The screenshot shows the configuration for a dispatcher task execution policy. The interface has three tabs: Options, Policy, and Jobs. The 'Policy' tab is active. The configuration is organized into several sections:

- Dispatcher task execution policy:**
 - Check intv.: 5 Seconds
 - Handle tasks
 - Evaluate relations
 - Evaluate approvals
 - Housekeeping actions
 - Housekeeping intv.: 30 Seconds
- Run jobs:**
 - Run jobs
- Java runtime engine:**
 - Run provisioning jobs
 - Run regular jobs
 - Log level: Error
 - Stack trace: Topmost entry
 - Java options: (empty text box)
- Windows runtime engine:**
 - Run provisioning jobs
 - Run regular jobs

The "Check intv" is set to 5 seconds for both dispatchers.

The dispatcher is configured to process tasks and evaluate relations and approvals.

The dispatcher should handle jobs for the Java runtime engine.

Using the Windows runtime engine

If your system requires use of the Windows runtime engine, and your dispatcher is running on a Unix server, a separate dispatcher must handle these jobs.

This dispatcher configuration could be like this:

The screenshot shows the configuration interface for a dispatcher. It has three tabs: 'Options', 'Policy', and 'Jobs'. The 'Jobs' tab is active. The main area is titled 'Dispatcher task execution policy'. It contains the following settings:

- Check intv.: 5 Seconds
- Handle tasks
- Evaluate relations
- Evaluate approvals
- Housekeeping actions
- Housekeeping intv.: 30 Seconds
- Run jobs

Under the 'Java runtime engine' section:

- Run provisioning jobs
- Run regular jobs
- Log level: Error
- Stack trace: Topmost entry
- Java options: (empty text box)

Under the 'Windows runtime engine' section:

- Run provisioning jobs
- Run regular jobs

The dispatcher is configured only to run jobs for the Windows runtime engine.

Performance considerations

In a production environment it is important to configure the dispatcher correctly to optimize performance of the system.

These are some areas to consider:

- Controlling the number of runtime engines started by the dispatchers.
- Controlling job looping of runtime engines

Controlling the number of runtime engines

The number of concurrent runtime engines for each dispatcher and for the system in total must be tuned. If you do not limit the number of concurrent runtime engines, you may end up in a thrashing situation. In this case, the dispatcher will find that there are jobs to be executed, and attempt to start the required number of runtime engines. Due to system load, it may take some time before the runtime engines actually get the jobs and the dispatcher will find that there are still jobs to be executed, and start even more runtime engines, making the situation worse.

The "Check intv" specifies the frequency with which the dispatcher should check for jobs to execute. As a general rule of thumb, the "Check intv" should be the number of dispatchers running in the configuration multiplied with 5 seconds. The factor of 5 seconds is recommended based on experience with running the dispatcher in production environments.

If you for example have 3 dispatchers, the "Check intv" should be set to 15 seconds for each of them, assuming that they all are configured for the same jobs.

There are three parameters that determine how many runtime engines are started at each "Check intv":

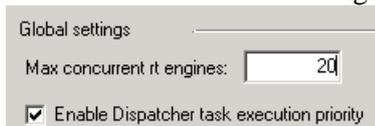
- The global setting "Max concurrent rt engines" specifies how many runtime engines are allowed on the system in total. If this is 0, there is no global limit.
- The setting "Max concurrent rt engines" for the dispatcher controls how many runtime engines are allowed for this dispatcher. If this is 0, there is no limit.
- The setting "Max rt engines to start" specifies how many runtime engines are started at once.

How many runtime engines are started for each "Check intv" will be calculated using these numbers.

An example

A system is configured as follows:

- Global "Max concurrent rt engines" is set to 20.

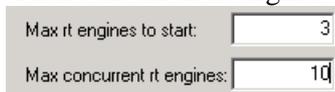


Global settings

Max concurrent rt engines:

Enable Dispatcher task execution priority

- "Max concurrent rt engines" for the dispatcher is set to 10.



Max rt engines to start:

Max concurrent rt engines:

- "Max rt engines to start" is set to 3.

The current situation at the "Check intv" time is:

- Total number of runtime engines is 12.
- The number of runtime engines for the current dispatcher is 8.

This gives the following calculation:

- The difference between the current number of runtime engines (12) and the total number of concurrent runtime engines (20), gives 8.
- The difference between the number of concurrent runtime engines for this dispatcher (8) and the number of max concurrent runtime engines for the dispatcher (10), gives 2.
- The number of runtime engines to be started at once is 3.

The lowest of these three numbers is 2 – resulting in two runtime engines to be started.

Calculating concurrent runtime engines

When calculating the number of concurrent runtime engines, only runtime engines running provisioning jobs are counted. This means that runtime engines running scheduled (regular) jobs always will be started, regardless of how many runtime engines already are running.

In a normal system it is assumed that there are more jobs defined for Java runtime engines than for the Windows runtime engine. Thus, the Java runtime engine is given priority when calculating how many runtime engines to start.

Given a total of 10 concurrent runtime engines for a dispatcher that is configured to run both Java and Windows runtime engines, the dispatcher will never start more than 5 Windows runtime engines (50%) and never more than 9 Java runtime engines (90%). This ensures that there will always be possible to start a Windows runtime engine, but that the Windows runtime engines will never occupy more than half of the available runtime engines.

This can be controlled by the advanced properties of the dispatcher, but it is not recommended to change these values unless you have a good reason to do so.

Controlling job looping of runtime engines

The dispatcher is normally installed as a service and started automatically by the system. Depending on the configuration of the dispatcher, it will regularly perform the specified operations. These intervals are controlled by "Check intv" and "Housekeeping intv" on the "Policy" tab of the dispatcher properties.

If the dispatcher is configured to run jobs, the runtime engine can process several jobs before it restarts. This is controlled by "Max loops for rt engine" on the "Options" tab of the dispatcher properties. The runtime engine will loop executing jobs the number of times specified here. This will improve performance, as the runtime engine does not have to restart to get the next job.

However, if there are memory leaks in any scripts or other custom code, they which will accumulate over time. Thus, this number should not be too high, to allow the runtime engine to restart and reinitialize.