Integrating IBM Security and SAP Solutions

- SAP business solutions, security, and the user and role management concepts
- IBM Security identity and access management integration
- Use cases and best practices

Anthony Ferguson
Vaughan Harper
David Moore
Zoran Radenkovic
Guy Redding
John Robinson
Sascha Schefenacker
Franz Wolfhagen

Axel Buecker
Ivy Chiu
Kenny Chow
Ingo Dressler

ibm.com/redbooks
Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (February 2012)

This publication discusses several software applications from IBM and SAP. The applicable versions are mentioned in the individual chapters of this publication.

© Copyright International Business Machines Corporation 2012. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Notices ................................................................. xi
Trademarks ........................................................... xii

Preface ................................................................. xiii
The team who wrote this book ........................................ xiii
Now you can become a published author, too! ....................... xvi
Comments welcome .................................................. xvii
Stay connected to IBM Redbooks .................................... xvii

Part 1. Business context and SAP solution overview ...................... 1

Chapter 1. Business context for SAP security integration ............... 3
1.1 Drivers that influence security .................................... 5
  1.1.1 Business drivers that influence security ..................... 5
  1.1.2 IT drivers that influence security ........................... 8
1.2 IBM Security Framework ......................................... 10
  1.2.1 Security Governance, Risk Management, and Compliance .. 12
  1.2.2 People and Identity domain ............................... 13
1.3 IBM Security Blueprint ........................................... 15
1.4 Security challenge ............................................... 18
1.5 IBM Reference Architecture for SAP solutions .................. 20
1.6 IBM implementation approach for SAP authorization ............ 22
1.7 Conclusion ......................................................... 26

Chapter 2. Introduction to SAP solutions and security technology ..... 29
2.1 SAP systems and applications .................................... 30
  2.1.1 SAP Business Suite ....................................... 30
  2.1.2 SAP NetWeaver ........................................... 32
2.2 SAP security and SAP user and role management concept ........ 34
  2.2.1 SAP NetWeaver AS ABAP User Repository ............... 37
  2.2.2 SAP NetWeaver AS Java User Repository: UME .......... 38
  2.2.3 SAP Central User Administration .......................... 38
  2.2.4 SAP NetWeaver Identity Management ...................... 40
  2.2.5 SAP BusinessObjects governance, risk, and compliance .. 41
2.3 SAP user management integration options and interfaces .......... 41
  2.3.1 Business Application Programming Interfaces (BAPI) .... 43
  2.3.2 Remote Function Calls (RFC) ............................ 44
  2.3.3 Synchronous versus asynchronous integration ............. 44
2.4 SAP access management integration options ........................ 46
2.4.1 SAP logon ticket ........................................... 46
2.4.2 Secure Network Communication .......................... 47
2.4.3 Digital certificates ........................................ 48
2.4.4 Security Assertion Markup Language ................... 49
2.4.5 Kerberos .................................................. 53
2.4.6 Single sign-on technologies ............................... 54
2.5 Conclusion ..................................................... 54

Part 2. Identity management integration .......................... 55

Chapter 3. IBM Security identity management offerings .......... 57
3.1 IBM Tivoli Identity Manager .................................. 58
  3.1.1 IBM Tivoli Identity Manager concept .................... 59
  3.1.2 Tivoli Identity Manager adapter concept ................. 59
  3.1.3 Adapter operations ...................................... 60
  3.1.4 Tivoli Identity Manager integration with SAP solutions . 61
  3.1.5 SAP user provisioning with IBM Tivoli Identity Manager . 61
3.2 IBM Tivoli Directory Integrator ................................ 70
  3.2.1 Tivoli Directory Integrator adapter framework ........... 73
  3.2.2 Tivoli Directory Integrator integrations with SAP solutions . 73
3.3 IBM Tivoli Directory Server .................................. 74
  3.3.1 Identity data foundation for security management applications . 75
  3.3.2 Tivoli Directory Server integration with SAP solutions ...... 75
3.4 Conclusion ..................................................... 76

Chapter 4. IBM Tivoli Identity Manager ............................ 77
4.1 Tivoli Identity Manager Adapter for SAP NetWeaver ......... 78
  4.1.1 Architecture overview ................................... 79
  4.1.2 Adapter implementation, customization, and extension .......... 80
4.2 Tivoli Identity Manager Adapter for SAP ABAP ................ 101
4.3 Tivoli Identity Manager Adapter for SAP NetWeaver AS Java .... 102
4.4 Tivoli Identity Manager Adapter for SAP Governance, Risk, and Compliance Access Control ......................... 103
4.5 Conclusion ..................................................... 111

Chapter 5. IBM Tivoli Directory Integrator ......................... 113
5.1 Function component for SAP NetWeaver AS ABAP .......... 114
5.2 User Registry Connector for SAP NetWeaver AS ABAP .......... 117
5.3 HR/Business Object Repository Connector for SAP NetWeaver AS ABAP .... 119
5.4 IDOC Connector for SAP ERP and SAP NetWeaver AS ABAP .... 121
5.5 Conclusion ..................................................... 123

Chapter 6. IBM Tivoli Directory Server ............................ 125
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1  Integration with SAP NetWeaver AS ABAP</td>
<td>126</td>
</tr>
<tr>
<td>6.1.1 Definition of attribute mappings</td>
<td>126</td>
</tr>
<tr>
<td>6.1.2 Definition of required schema extension</td>
<td>126</td>
</tr>
<tr>
<td>6.1.3 SAP NetWeaver AS ABAP LDAP configuration</td>
<td>126</td>
</tr>
<tr>
<td>6.1.4 Directory synchronization</td>
<td>129</td>
</tr>
<tr>
<td>6.2  Integration with SAP NetWeaver AS Java</td>
<td>130</td>
</tr>
<tr>
<td>6.2.1 General overview of the basic component LDAP interface</td>
<td>130</td>
</tr>
<tr>
<td>6.2.2 Tivoli Directory Server data source configuration for UME</td>
<td>131</td>
</tr>
<tr>
<td>6.3  Conclusion</td>
<td>141</td>
</tr>
<tr>
<td>7.1  SAP HR driven identity feed</td>
<td>144</td>
</tr>
<tr>
<td>7.1.1 SAP IDOC configuration</td>
<td>146</td>
</tr>
<tr>
<td>7.1.2 Configuring the Tivoli Directory Integrator AssemblyLine</td>
<td>147</td>
</tr>
<tr>
<td>7.1.3 Position-based user provisioning</td>
<td>148</td>
</tr>
<tr>
<td>7.2  Tivoli Directory Server on z/OS and SAP solutions</td>
<td>149</td>
</tr>
<tr>
<td>7.2.1 Why choose Tivoli Directory Server for z/OS over a distributed platform</td>
<td>149</td>
</tr>
<tr>
<td>7.2.2 Tivoli Directory Server for z/OS configuration</td>
<td>150</td>
</tr>
<tr>
<td>7.3  Tivoli Directory Server as shared user repository</td>
<td>157</td>
</tr>
<tr>
<td>7.3.1 Configuring SAP NetWeaver AS Java UME for IBM Tivoli Directory Server</td>
<td>158</td>
</tr>
<tr>
<td>7.3.2 Configuring Tivoli Directory Server write authority</td>
<td>161</td>
</tr>
<tr>
<td>7.4  Tivoli Identity Manager sample scenarios and use cases</td>
<td>162</td>
</tr>
<tr>
<td>7.4.1 Stand-alone SAP ABAP target server</td>
<td>163</td>
</tr>
<tr>
<td>7.4.2 CUA target server</td>
<td>163</td>
</tr>
<tr>
<td>7.4.3 Multiple SAP ABAP stand-alone target servers</td>
<td>165</td>
</tr>
<tr>
<td>7.4.4 Stand-alone ABAP target with HR modules</td>
<td>166</td>
</tr>
<tr>
<td>7.4.5 CUA target environment with HR modules</td>
<td>167</td>
</tr>
<tr>
<td>7.4.6 Single sign-on password management in a stand-alone setup</td>
<td>168</td>
</tr>
<tr>
<td>7.4.7 Single sign-on password management with a CUA target</td>
<td>169</td>
</tr>
<tr>
<td>7.4.8 Account locking extension for CUA and non-CUA setup</td>
<td>170</td>
</tr>
<tr>
<td>7.5  Tivoli Identity Manager best practices</td>
<td>171</td>
</tr>
<tr>
<td>7.5.1 Deployment of Tivoli Directory Integrator</td>
<td>171</td>
</tr>
<tr>
<td>7.5.2 Performance issues</td>
<td>173</td>
</tr>
<tr>
<td>7.5.3 High availability</td>
<td>174</td>
</tr>
<tr>
<td>7.5.4 A global SAP infrastructure</td>
<td>176</td>
</tr>
<tr>
<td>7.5.5 SAP Secure Network Communications (SNC)</td>
<td>179</td>
</tr>
<tr>
<td>7.5.6 SAP version mixture</td>
<td>179</td>
</tr>
<tr>
<td>7.5.7 Unicode and non-Unicode support</td>
<td>179</td>
</tr>
<tr>
<td>7.5.8 SAP message server</td>
<td>180</td>
</tr>
<tr>
<td>7.6  Conclusion</td>
<td>180</td>
</tr>
</tbody>
</table>
Part 3. Access management integration ............................................. 181

Chapter 8. IBM Security access management offerings ...................... 183
8.1 Tivoli Access Manager for Enterprise Single Sign-On .................. 186
  8.1.1 AccessProfiles ......................................................... 186
  8.1.2 Integration with SAP solutions ............................................. 187
8.2 Tivoli Access Manager for e-business ............................................ 188
  8.2.1 Components ............................................................... 188
  8.2.2 Integration with SAP solutions ......................................... 192
8.3 Tivoli Federated Identity Manager .................................................. 193
  8.3.1 Federated Identity Management functionality .......................... 193
  8.3.2 Integration with SAP solutions ......................................... 195
8.4 Conclusion ................................................................. 195

Chapter 9. IBM Tivoli Access Manager for Enterprise Single Sign-on ...... 197
9.1 Single sign-on considerations ................................................. 198
9.2 System name display requirements ........................................... 198
9.3 User education ........................................................................ 199
9.4 Implementation process overview .............................................. 200
9.5 Implementing Tivoli Access Manager for Enterprise Single Sign-On ... 200
9.6 How the AccessProfile works ...................................................... 201
  9.6.1 Application process control .................................................. 201
  9.6.2 Number of capture or injection fields .................................... 201
  9.6.3 Single sign-on to SAP applications using SAP Logon ................. 202
  9.6.4 Single Sign-On to SAP applications using SAP Shortcut ............. 204
  9.6.5 AccessProfile language support ............................................ 205
  9.6.6 AccessProfile environment and default settings ....................... 206
  9.6.7 Setting up the registry for SAP Logon .................................... 208
  9.6.8 Capturing credentials when signing in to SAP applications ........... 210
  9.6.9 Generating random password during password change ................ 211
  9.6.10 Authentication service names ............................................ 213
9.7 Web single sign-on to SAP GUI for HTML ..................................... 215
  9.7.1 SAP GUI for HTML ........................................................... 216
  9.7.2 Web Single Sign-On using Internet Explorer browser .................. 217
  9.7.3 Authentication service name for Internet Explorer profile ............... 218
  9.7.4 Web Single Sign-On using Firefox browser ................................ 218
9.8 Conclusion ............................................................................. 219

Chapter 10. IBM Tivoli Access Manager for e-business ......................... 221
10.1 Integration with SAP NetWeaver AS ABAP .................................... 222
  10.1.1 WebSEAL junctions to SAP NetWeaver AS ABAP .................... 224
  10.1.2 Configuring Tivoli Access Manager WebSEAL options ............... 226
  10.1.3 Configuring SAP NetWeaver AS ABAP ................................ 226
  10.1.4 Testing the integration ...................................................... 227
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- Active Memory™
- AIX®
- AppScan®
- AS/400®
- Cognos®
- DataPower®
- DB2 Connect™
- DB2®
- Domino®
- DRDA®
- DS8000®
- Global Business Services®
- Global Technology Services®
- Guardium®
- HACMP™
- i5/OS®
- IBM®
- InfoSphere®
- Lotus Notes®
- Lotus®
- Micro-Partitioning®
- Notes®
- Optim™
- OS/400®
- Power Architecture®
- Power Systems™
- POWER7®
- PowerHA®
- PowerVM®
- POWER®
- RACF®
- Rational®
- Redbooks®
- Redbooks (logo)®
- Service Request Manager®
- System p®
- System z®
- Tivoli®
- WebSphere®
- z/OS®

The following terms are trademarks of other companies:

- Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

- Itanium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

- Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

- Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

- UNIX is a registered trademark of The Open Group in the United States and other countries.

- ABAP, BAPI, SAP NetWeaver, SAP R/3, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

- Other company, product, or service names may be trademarks or service marks of others.
Preface

Many large and medium-sized organizations have made strategic investments in the SAP NetWeaver technology platform as their primary application platform. In fact, SAP software is used to manage many core business processes and data. As a result, it is critical for all organizations to manage the life cycle of user access to the SAP applications while adhering to security and risk compliance requirements.

In this IBM® Redbooks® publication, we discuss the integration points into SAP solutions that are supported by the IBM Security access and identity management product capabilities. IBM Security software offers a range of identity management (IdM) adapters and access management components for SAP solutions that are available with IBM Tivoli® Identity Manager, IBM Tivoli Directory Integrator, IBM Tivoli Directory Server, IBM Access Manager for e-business, IBM Tivoli Access Manager for Enterprise Single Sign-On, and IBM Tivoli Federated Identity Manager.

This book is a valuable resource for security officers, consultants, administrators, and architects who want to understand and implement an identity management solution for an SAP environment.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Axel Buecker is a Certified Consulting Software IT Specialist at the ITSO, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of software security architecture and network computing technologies. He holds a degree in computer science from the University of Bremen, Germany. He has 25 years of experience in a variety of areas related to workstation and systems management, network computing, and e-business solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.

Ivy Chiu is a Software Developer at the IBM Australian Development Lab (ADL), Gold Coast. She has four years of experience specializing in the testing and development of IBM Tivoli Identity Manager adapters. Her experiences cover testing of the Tivoli Identity Manager Adapter for SAP NetWeaver, Tivoli Identity
Manager Adapter for SAP ABAP, and Tivoli Identity Manager Adapter for SAP GRC. She has a degree in information technology from the Queensland University of Technology, Australia.

**Kenny Chow** is a Software Engineer at the IBM Singapore Software Lab. He has over seven years of software development experience with a focus on systems integration and communication protocols. Since he joined IBM, he has been working on the integrations for Tivoli Access Manager for Enterprise Single Sign-On and Tivoli Identity Manager. He holds a degree in computer science and engineering from the State University of New York at Buffalo.

**Ingo Dressler** is a Certified Security Consultant at the IBM SAP International Competence Center, Walldorf, Germany. He has 16 years of IT experience, including 10 years of experience in the information security field. He has a degree in computer science from the University of Cooperative Education of Dresden. His areas of expertise include SOA security and user-centric identity and access management. By joining the research and development division of IBM Germany in 2004, he specialized on the security integration aspects of SAP technology-based architectures.

**Anthony Ferguson** is a Software Developer at the IBM Australian Development Lab (ADL), Gold Coast. He has seven years of software development and support experience. He started his career as a Developer in the IBM Global Security Kit component team and then moved into the Tivoli Access Manager for e-business Level-3 support team. He currently works as a Level-3 Support Lead for the IBM Tivoli Integration Factory. He has a degree in information technology from the University of Central Queensland, Australia.

**Vaughan Harper** is an IBM Security Architect with 25 years of experience in the IT industry, the last 15 years in IT Security. He spent four years as the EMEA Technical Evangelist for IBM Tivoli Access Manager for e-business, and for the last eight years he has supported Tivoli Security products in the UK. Prior to joining the IBM Software Group he consulted, implemented, and supported IBM cryptographic and network security products. Before that he held various software development and system design jobs in the UK and the US. He has a master's degree in engineering from the University of Cambridge.

**David Moore** is a Software Developer at the IBM Tivoli Security Development Lab, Gold Coast, Australia. He has 14 years of software development experience. David's primary areas of expertise are middleware, security, identity, and governance, risk, and compliance (GRC) integration with a particular focus on SAP technologies. David has a computer science degree from Griffith University.

**Zoran Radenkovic** is a member of the MEA Tivoli Technical Sales team located in United Arab Emirates. Before that Zoran was part of IBM Tivoli Integration Factory located in Australia, as Technical Lead for the Tivoli Identity Manager.
Adapter development team. Zoran has more than 20 years of experience in software development. Prior to joining IBM he developed software for cryptography, real-time operating systems, embedded systems, cryptography accelerators and HSM, distributed files systems, internet search engines, voice-compressions, networking, and ASIC design tools.

**Guy Redding** is a Software Engineer at the Gold Coast IBM Australia Development Labs in Australia. He joined IBM in January 2010 and is currently working in the Tivoli Security Integration Factory, primarily focused on SAP product integration with IBM Tivoli Identity Manager. He has a background in academic and industrial IT research and has a PhD in business process management from Queensland University of Technology, which was awarded in 2009.

**John Robinson** is a Senior Software Engineering Manager at the IBM ADL Gold Coast site in Australia. He manages development of several Tivoli security products in the Access and Identity space. He has bachelor’s and master’s degrees in electrical engineering and computer systems engineering. Prior to his management roles, he lead the Tivoli Security Integration Factory team for eight years. He has over 20 years of experience with software engineering, specializing in security software architecture, design, and development. He has been involved with Tivoli Access Manager for Enterprise Single Sign-On for the past three years, and has developed many of the AccessProfiles currently in use.

**Sascha Schefenacker** is an IT Architect for IBM Software Group Germany. He joined IBM in 2000. In the last eight years, his work has been strongly SAP-specific. He is currently part of the Boeblingen laboratories. The team is responsible for building IBM WebSphere® architecture for use with SAP software, including customer assignments and proof of concepts. His main responsibility is to evaluate, design, and architect areas where IBM can be used in conjunction with SAP software. His additional areas of focus are WebSphere Portlet Factory, IBM Rational® Application Developer, and Portal-based projects. He is an IBM Certified Solution Designer for service-oriented architectures, specialized to combine SAP NetWeaver solutions and the IBM WebSphere middleware process platform. He has a Bachelor of Science degree in information technology, project management, and e-business from the University of Cooperative Education, Stuttgart.

**Franz Wolfhagen** is Certified IT Specialist working in IBM Global Technology Services® in Denmark. Franz has been employed by IBM since 1983 and has worked with System Management since 1996. Since 2002 Franz has specialized in Identity and Access Management and has headed several large implementations in the Nordic area. Franz is considered to be a SME on IBM Tivoli Identity Manager in the Nordic area and has also presented at the Tivoli Technical Conferences in Europe.
Thanks to the following people for their contributions to this project:

Keith Barton, Allan Coulter, Simon Canning, Wolfgang Epting, Mike Hahn, Caecilie Hampel, Thomas Hanicke, Reinhard Heite, Sven Herschel, Jack Inniss, Klaus Jaeck, Gunter Jahn, Ashok Kallarakkal, Robert Kennedy, David Mackenzie, Brian Matthiesen, Manfred Paessler, Katharina Probst, Johannes Schuetzner, Frank Thurau, Peter Tuton, Guenter Waller

IBM

Frank Buchholz, Martin Raeple
SAP AG

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html
Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks

- Send your comments in an email to:
  redbooks@us.ibm.com

- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:
  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:
  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

- Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
Business context and SAP solution overview

In this first part of this book we describe the business aspects for an identity and access management life cycle when using an SAP enterprise resource planning (ERP) system. We briefly introduce the various SAP offerings and the concepts for security integration.
Business context for SAP security integration

Many large organizations have chosen to make strategic investments using SAP software to manage many core business processes and data. As a result, it is critical for those organizations to manage the identity and access management life cycle, not only to their SAP applications, but to their entire IT estate, while adhering to security and risk compliance requirements.

On the other side, there are enormous business transformation activities in the corporate world that drive an ever-increasing process integration of customers, business partners, and vendors. Higher integration requires increased protection of enterprise and client information against unauthorized access, intermediation, and manipulation.

At the same time, as the world becomes more interconnected and as the paradigm of the organization shifts from multinational corporation to globally integrated enterprise, businesses can no longer afford to operate as islands. More than ever, they need to find ways to connect both internally and externally.

This might be a challenge for many companies who run their SAP systems separately from their enterprise service and security management procedures, despite the fact that SAP and ERP systems are more and more connected to the internet. That is a result of IT progression where technology has allowed work to move beyond physical locations and brand offices. It now exists on virtual
platforms, across dispersed teams and players. But the more data and information that organizations make available to their internal audience, the greater the risk.

The only chance to handle all the risks is to get a holistic view of all the associated systems and procedures and also integrate SAP solutions into the enterprise security management dashboard.

This is where IBM Security solutions come into play. IBM Security solutions have been recognized as best of breed by key industry analysts. For SAP customers, there are tremendous benefits due to the IBM efforts in closely integrating and extending the SAP security capabilities using these technologies.

Key capabilities include these:

- Provisioning users across multiple SAP and non-SAP systems and applications.
- Extending SAP security to prevent unauthorized access to information and applications.
- Automatic alerting of suspicious activities and capture of audit information for regulatory compliance.

Let us take a closer look at the IBM Security core foundation around the IBM Security Framework.
1.1 Drivers that influence security

Most of today's projects are driven by both business and IT drivers, although we can probably agree that business drivers are almost always the initiating factor. Let us take a closer look at these influencing factors:

- **Business drivers**
  
  Business drivers measure value, risk, and economic costs that influence their approach to IT security. Value drivers determine the worth of assets of the system to the business and of the business itself. Risk drivers involve compliance, corporate structure, corporate image, and the risk tolerance of the company. Economic drivers determine productivity impact, competitive advantage, and system cost.

- **IT drivers**
  
  IT drivers represent operational constraints in the general IT environment. For example, the complexity of a system, including its environment, that is exposed to internal and external threats presents risks that the organization must address.

Business drivers also represent issues and consequences of significance to the stakeholders of the managed business system. This set of drivers might vary from industry to industry, from organization to organization in the same industry, and even between various business applications in an organization.

IT drivers represent technical considerations that affect the trustworthiness of the IT environment and likely the managed business systems as a whole. IT drivers are universal and must be considered within the context of the business drivers in all efforts. The combination of business and IT drivers represents the key initiatives for security management.

1.1.1 Business drivers that influence security

Business drivers represent a relationship between the IT organization and the rest of the business. They refer to business values that must be supported by the IT security infrastructure.

**Correct and reliable operation**

Correct and reliable operation is the degree to which the business must be accurate and consistent in its operation. Correct operation means that the operations perform the proper response or function with no errors. Reliable means that the same result occurs all the time. Any IT system must consistently provide stakeholders with the expected results.
Security events and incidents might impact the correct and reliable operation of these business processes. It might also affect the underlying IT infrastructure or upstream and downstream business processes. The consequences of a defective service (incorrect or varying results over time) might be significant to the consumer of the service, and therefore to the provider of the service.

**Service-level agreements**
This driver applies to circumstances where security threats and threat agents can impact an organization’s ability to conduct business. Service-level agreements (SLAs) incorporate acceptable conditions of operation within an organization. SLAs might vary from business system to business system or application to application. Availability of systems, data, and processes is a condition commonly referenced within SLAs.

**IT asset value**
From a business perspective, the IT asset value is directly related to the value of the business transactions that it supports. These might be tangible or intangible. For an e-retailer, these are tangible assets. For a financial services company, the asset might be the client information or other data used in transactions of the system.

**Protection of the business asset value or brand image**
This driver captures the firm’s desire to protect its image. The loss of good will from a security incident or attack has a direct consequence to the business. Therefore, the security measures are likely to be proportional to the consequence. When the desire to avoid negative publicity increases, upon encountering a security breach, the stipulation for this driver becomes stronger.

**Legal and regulatory compliance**
Legal and regulatory compliance refers to the externally imposed conditions on the transactions in the business system and the company. This includes the rules and policies imposed by regulatory and government agencies. Civil, criminal liability, or regulatory penalty from a security incident or attack have a negative impact on the business. Therefore, the amount of regulation and steps to ensure compliance should be factored into this driver. This includes privacy issues, the ability to prove the transaction initiator, and proving compliance.

An implemented log management system can tell who did what, where, and when. Log management, therefore, is an integral part of an IT security compliance management system. For the retention period of the logs, it is ensured that the necessary information is available and can be analyzed or interpreted to a level that can help management to better investigate security incidents or comply with external regulation or laws. Compliance is a key
business driver today, and log management should be a part of every IT security compliance management solution. But it can also be implemented alone as an initial step towards a larger IT security compliance initiative. As already mentioned, many international standards and regulatory controls require logging to be enabled and implemented. Also, these logs must be analyzed periodically and stored for a specific period of time, depending on the particular standard or regulatory control.

**Contractual obligation**
Security measures for an IT system are likely to be proportional to the consequences encountered when the business encounters contractual liability from a security attack. Depending on the structure and terms of the contract, the consequence might lead to financial loss or liability. For example, when security incidents are encountered, the business might be unable to fulfill its contractual obligations of providing goods or services.

**Financial loss and liability**
Direct or indirect financial loss is a consequence to the business as a result of a security incident. Direct loss might include theft of an asset, theft of a service, or fraud. Indirect loss might include loss based on civil or criminal court ruling, loss of good will, or re-prioritized budget allocation. This driver identifies the fact that security measures for an IT system are likely to be in proportion to these consequences.

**Critical infrastructure**
This driver applies where security threats or threat agents can have a major impact on services or resources that are common to, or shared among, a community of businesses, the population at large, or both. Examples include telecommunications, electrical power, transportation systems, computing, and so on. The loss of critical infrastructure by its provider might have a ripple effect, causing secondary losses and driving security decisions for those affected. An important part of risk analysis is identifying critical infrastructure.

**Safety and survival**
This driver applies where security threats and threat agents can have a major impact on aspects of human life, government function, and socio-economic systems. Examples of processes to be considered for safety and survival impact include continuity of critical infrastructure, medical systems, life support, or other high-impact or time-dependent process.
1.1.2 IT drivers that influence security

IT drivers comprise the second group of key security initiatives. These are considered universal drivers that must be considered in every modern IT solution in a manner commensurate with the risks and consequences of a related failure or incident.

**Internal threats and threat agents**

Security-related failures and incidents are caused by threats or threat agents found within the physical and logical boundaries of the organization or enterprise that operates and controls the IT system. These threats and threat agents might be associated with technology or people.

An example of an internal threat is a poorly designed system that does not have the appropriate controls. An example of an internal threat agent is a person who would use his ability to access the IT system or influence business or management processes to carry out a malicious activity.

**External threats and threat agents**

Security-related failures and incidents are caused by threats or threat agents found outside the physical and logical boundaries of the organization or enterprise that operates and controls the IT system. These threats and threat agents are also associated with technology or people. They seek to either penetrate the logical or physical boundary to become internal threats or threat agents, or to influence business or management processes from outside the logical or physical boundary.

Examples of external threats are single points of failure for one or more business or management processes that are outside the enterprise boundary, such as a power system grid or a network connection, or a computer virus or worm that penetrates the physical or logical network boundary. An example of an external threat agent is a hacker, or someone who has gained the ability to act as an insider using personal electronic credentials or identifying information.
IT service management commitments
This driver identifies the fact that failure to manage the operation of the IT system might result in security exposures to the business. This driver can be divided into two categories, IT service delivery and IT service support:

- Service delivery commitments
  The failure of the IT system to meet its metrics for managing itself can be viewed as a security exposure to both business and management processes.
  An example of security exposure for service delivery is when IT operations processes cannot respond to critical events in a timely manner. Another is when IT resilience processes cannot recover from a denial of service attack in a timely manner, resulting in a loss of capacity or response time for business processes.

- Service support commitments
  The failure of the business or IT management system to meet its service-level agreements can be viewed as a security exposure to business or management processes.
  An example of security exposure for service support is a situation in which the customer relationship processes do not add, modify, or remove users from access control lists in a timely manner.

IT environment complexity
The complexity of the IT environment might contribute to the security or insecurity of the IT system. The IT environment reflects the infrastructure on which the business system will be placed. For example, any IT environment that is connected to the intranet or extranet is exposed to internal or external threats or threat agents and requires specific security responses. A stand-alone facility for our system represents the lowest complexity. A hosting facility with other systems and other firms represents a more complex environment. An environment with a larger number of systems, varied network access paths, or a complex architecture is a complex IT environment.

Business environment complexity
Because most businesses rely on IT, most business environments are an interconnected set of businesses, each with its own complex IT environment, business processes, and IT management processes. This complexity might contribute to the security or insecurity of the IT system.

Audit and traceability
This driver identifies the need for the IT system to support an audit of information contained within the system, whether it is associated with management data or business data.
IT vulnerabilities: Configuration
Configuration vulnerabilities are potentially present in every IT system, providing an opening to a potential attack based on the system and how it is designed and set up.

IT vulnerabilities: Flaws
Software flaws potentially exist in every IT system. These flaws represent vulnerabilities that were not detected and are not evident in the design documents. As such, they are an unexpected deviation from what was designed. An example is a defect in an operating system or application that is discovered after implementation.

IT vulnerabilities: Exploits
The basic design of software in any IT system might be exploited by threats or threat agents as a part of an attack on the IT system, the business, or the management processes. This might include the use of a function within a system in a way to compromise the system or underlying data. While certain people might define an exploit as both the flaw and the method, we treat them separately because an exploit might involve using normal functions as designed in an unusual manner to attack the system. The exploits can also be viewed as the openings or avenues that an attacker can use.

Now it is time for us to introduce the IBM Security Framework, which focuses on the what, not the how. It can help you translate your requirements into coarse-grained business solutions, not into specific IT components or IT services.

1.2 IBM Security Framework

Today’s business leaders are expected to manage risk in their areas of responsibility in the same way that CFOs manage risks in their domains. Security risks and the potential impact on IT need to be communicated to executive peers in business terms. Additionally, they need to align IT security controls with their business processes, monitor and quantify IT risk in business terms, and dynamically drive business-level insight at the executive level. Finally, business leaders need to manage risk and orchestrate security operations in a way that enforces compliance and optimizes business results.

As an organization secures its business processes, a business-driven approach needs to become the guiding influence for ensuring that all the security domains work together in a holistic and synergistic manner, in alignment with the overarching business objectives. Otherwise, the organization’s risk stance becomes vulnerable due to misalignment of priorities between IT and the
business strategy. Using a standards-based approach to map business drivers to IT security domains is often difficult and is often an afterthought.

IBM created a comprehensive IT security framework (Figure 1-1) that can help ensure that every necessary IT security domain is properly addressed when using a holistic approach to business-driven security.

![IBM Security Framework](image)

*Figure 1-1  The IBM Security Framework*

IBM provides the full breadth and depth of solutions and services that can enable organizations to take this business-driven, secure-by-design approach to security in alignment with the IBM Security Framework. Comprehensive professional services, managed services, and hardware and software offerings are available from IBM to support your efforts in addressing the various security domains covered by the IBM Security Framework.
1.2.1 Security Governance, Risk Management, and Compliance

**Naming similarity:** You might find similar wording by looking at the SAP product portfolio. There is the SAP BusinessObjects suite of products for governance, risk, and compliance management that mainly addresses the automation of controls and risk mitigation for business applications, and those from SAP specifically. The SAP BusinessObjects GRC solutions address the CFO's needs for risk analysis and to balance the costs of risk avoidance with other opportunities, in addition to monitoring risks continuously. In this way the SAP BusinessObjects GRC solutions, such as for global trade services or access risk management, are complementary to the IBM Security Framework and integrate with the IBM security product portfolio.

Every organization needs to define and communicate the principles and policies that guide the business strategy and business operation. In addition, every organization must evaluate its business and operational risks, and develop an enterprise security plan to serve as a benchmark for the execution and validation of the security management activities that are appropriate for their organization.

These principles and policies, the enterprise security plan, and the surrounding quality improvement process represent the enterprise Security Governance, Risk Management, and Compliance model. Specifically, these are the requirements and the compliance criteria for the five security domains:

- **People and Identity**
  This domain covers aspects about how to ensure that the correct people have access to the correct assets at the correct time.

- **Data and Information**
  This domain covers aspects about how to protect critical data in transit or at rest across the organization.

- **Application and Process**
  This domain covers aspects about how to ensure application and business services security.

- **Network, Server, and Endpoint (IT infrastructure)**
  This domain covers aspects about how to stay ahead of emerging threats across IT system components.

- **Physical Infrastructure**
  This domain covers aspects about how to use the capability for digital controls to secure events—on people or things—in the physical space.
In the following section we take a closer look at the People and Identity domain. We focus on this domain because it is the driving factor for implementing a security policy management solution. If you want to learn more about the other IBM Security Framework domains, see the IBM Redpaper™ publication *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

### 1.2.2 People and Identity domain

Organizations need to protect the assets and services that serve the business and support the business operation. One aspect of protection is provided by *access control*. The ability to provide effective access control services is based on the ability to manage people and identity as defined by the enterprise’s security governance, risk, and compliance model.

**Note:** SAP systems and applications can play a pivotal role in this context. For example, SAP ERP applications or SAP Business Information Warehouse components are crucial for business processes and operations. They can be regarded as assets that need to be protected by proper access controls.

There are authoritative sources of identity information such as SAP ERP Human Capital Management (HCM), in addition to user interface, middleware, and security software from SAP that can be considered vital parts of the business processes throughout the identity life cycle.

The Security Governance, Risk Management, and Compliance model provides guidance about how identities are managed and how access control is to be conducted. Organizations register people and map them to identities. The relationships between people and organization are expressed in terms of role, rights, business policies, and rules. The ability to register people and describe their relationship with the enterprise is a key security enabler for the remaining security domains:

- Data and Information
- Applications and Process
- Network, Server, and Endpoint (IT infrastructure)
- Physical Infrastructure

Operationally, people acting in authorized roles in an organization or as part of an extended relationship are granted access to infrastructure, data, information, and services. At the same time, people acting in unauthorized roles are denied access to infrastructure, data, information, and services if they are acting outside of the business policies and agreements.
Within an identity system, people can be issued a credential. A credential can take any of several forms, including a physical identity card or logical token or user identifier. The trustworthiness or strength of the credential is an important aspect of business policy and risk management. The ability to effectively manage the life cycle of identity (that is, the creation, removal, and role changes for dynamic populations of workforce, customer, or user communities) is extremely important. For example, the life cycle of identities and credentials can be influenced by business cycles, employment cycles, customer relationship, agreement, business, or calendar events, and so on.

Identity systems need to be integrated with appropriate sets of access controls. Identity systems need to manage user roles, rights, and privileges across the IT infrastructure that might contain multiple technology architectures, or multiple identity and access control systems will be required to ensure that users have access to the correct assets and services.

Compliance for identity and access is often externally motivated compliance. For example, legislated privacy and evidence recording is a significant driver for implementation of comprehensive user provisioning and identity-related record keeping.

Figure 1-2 shows a summary and additional aspects to be addressed within the People and Identity domain.
After having addressed and mapped the IT security domain, People and Identity, into your business solutions, it is time to look at the component-oriented view of IT security in the IT Security Blueprint.

1.3 IBM Security Blueprint

The IBM Security Framework divides the area of business-oriented IT security into five domains. The next step is to break these down into further detail to work toward a common set of core security capabilities needed to help your organization securely achieve its business goals. These core security capabilities are called the IBM Security Blueprint.

The IBM Security Blueprint uses a product-agnostic and solution-agnostic approach to categorize and define security capabilities and services that are required to answer the business concerns in the IBM Security Framework.

The IBM Security Blueprint was created after researching many customer-related scenarios, focusing on how to build IT solutions. The intention of the blueprint is to support and assist in designing and deploying security solutions in your organization.

Building a specific solution requires a specific architecture, design, and implementation. The IBM Security Blueprint can help you evaluate these, but does not replace them. Using the IBM Security Blueprint in this way can provide a solid approach to considering the security capabilities in a particular architecture or solution.
IBM has chosen to use a high-level service-oriented perspective for the blueprint, based on the IBM service-oriented architecture approach. Services use and refine other services (for example, policy and access control components affect almost every other infrastructure component). See Figure 1-3 to better position and understand the IBM Security Blueprint.

The left portion of Figure 1-3 represents the IBM Security Framework, which describes and defines the security domains from a business perspective.

The middle portion in Figure 1-3 represents the IBM Security Blueprint, which describes the IT security management and IT security infrastructure capabilities needed in an organization. As discussed earlier, the IBM Security Blueprint describes these capabilities in product and vendor-neutral terms.

The right portion of Figure 1-3 represents the solution architecture views, which describe specific deployment guidance particular to a given IT environment. Solution architecture views provide details about specific products, solutions, and their interactions.

Figure 1-4 on page 17 highlights the components and subcomponents of the IBM Security Blueprint that have to be examined for every solution in the People and Identity security domain. Besides the Foundational Security Management, the IBM Security Blueprint enables you to determine the Security Services and Infrastructure components by reviewing the component catalogs for these Foundational Security Management services. Each of these components then can be assessed by determining whether each particular infrastructure component is
required to make a Foundational Security Management service functional so that it can address the issues or provide a prospected value associated with the particular business security domain, in this case, People and Identity.

We can see in Figure 1-4 that almost all infrastructure components might be required for a People and Identity security solution apart from Code and Images. The reason why those components are not included is that they are mostly covered by other domains of the IBM Security Framework.

1.4 Security challenge

SAP applications run at the core of an enterprise and require a particular focus on protection. Security concepts in context with ERP must protect at several layers, at the user level, within the communication level, at the application level, and last but not least at the database and data storage level.

SAP systems and applications have evolved from a single system to a complete solution architecture. The growing complexity of the SAP solution landscape (that is, two software stacks (ABAP and Java), SAP NetWeaver, and SAP Business Suite products) generates an increasing demand on flexibility, efficiency, and security.

At the same time, the dependency on the availability of IT staff gets more and more critical for successful IT operations. On top, the IT budget does not adjust to the growing needs, but mostly remains stable. The challenge is to improve service, reduce cost, and manage risk at the same time. The way to respond to this challenge adequately is by automation, while ensuring visibility and control.

Identity and access management requirements

From the administration perspective, SAP application security is mainly implemented on the technology side using the SAP NetWeaver technology platform. However, large organizations typically run IT infrastructures containing many other platforms, like databases, applications, and more. This approach can present an administrative overhead, because there is no central identity and access management (IAM) administration function, serving both SAP and non-SAP components. As a result:

- The benefits of industry standards are diminished and common integrations, such as support for multiple authentication devices and schemes, might not be uniformly available.

- The separate IAM administration “islands” for SAP and non-SAP infrastructures can result in separate and often incompatible access control policies, administration procedures, and even audit and compliance regimes.
In planning an identity and access management solution for an SAP solution landscape, the following characteristics need to be decided:

- The role definitions, to be provisioned through the identity management solution and enforced through the access management solution, should be aligned to the roles defined in the HR organization model.
- The distribution of user accounts and the scheme for synchronization of those accounts within the SAP clients.
- Responsibility for the definition and enforcement of separation of duties within the SAP environment.
- Access requirements for various classes of SAP users, such as power users via SAP GUI, casual users in SAP Portal, and business partners through federated access.
1.5 IBM Reference Architecture for SAP solutions

The deployment of IAM solutions in the context of using SAP software is part of the architectural guidelines as set out in IBM Reference Architecture gained from a range of implementation scenarios. The security aspect of the IBM Reference Architecture for SAP solutions in this section outlines the position on SAP security and how this can be enhanced by IBM Security solutions to provide a more comprehensive security offering (Figure 1-5).

In relation to identity management, in a number of SAP engagements Tivoli Identity Manager (TIM) has been deployed to provision user identity accounts to a combination of SAP user repositories, including these:

- Individual SAP ABAP user master records
- Multiple SAP ABAP child user master records, using SAP Central User Administration (CUA)
SAP NetWeaver Portal using the User Management Engine (UME)

Tivoli Access Manager for e-business (TAMeb), which can in turn share a user repository with SAP NetWeaver Portal via UME

Provisioning is often based on periodic, scheduled processing of HR extracts based on joiner/mover/leaver activity in the SAP ERP HCM organization model. Tivoli Identity Manager then carries out provisioning actions for SAP applications, using a role model synchronized to the HR organization model structure.

Where an organization has deployed SAP BusinessObjects GRC solutions, additional requirements can be placed on the provisioning solution:

- User account provisioning that enables auditing of account actions
- Attestation and approval of system access and role assignments at the IT operational or business level
- Enforcement of policies, at the time or point of account provisioning, designed to prevent occurrence of separation of duties (sod) and other risk violations

Tivoli Identity Manager turns provisioning requests into SAP BusinessObjects access control requests. The Tivoli Identity Manager adapter for SAP BusinessObjects access control requests uses the SAP BusinessObjects access control web service interface to submit the access request. This ensures that all provisioning actions are processed through the SAP BusinessObjects GRC access control functions, to prevent separation of duties conflicts.

In relation to access management, Tivoli Access Manager for e-business provides consistent access control and enforcement of security policies for all SAP resources and applications. This allows the following aspects:

- Access control for SAP resources and applications can be integrated into a single model, addressing platform operating systems, middleware, and non-SAP applications also.
- Primary authentication can leverage both the organization’s main user repository and any strong authentication technology (such as tokens, digital certificates, biometrics) to deliver a single, intuitive primary authentication for all users at the time of desktop login.

- Authentication and single sign-on (SSO) to SAP NetWeaver applications through the SAP Portal, using the SAP login ticket.

- Integration of desktop login to the SAP GUI through Tivoli Access Manager for Enterprise Single Sign-on.

- Integration to cloud, federated, and SOA environments with the use of Web Services Security, SAML protocols, and Tivoli Federated Identity Manager.
Before an implementation project with this functionality can be approached, the general SAP authorization concept needs to be modeled as a foundation for the organization. IBM Global Business Services® has developed a methodology that can be used for that purpose. In the following section we provide a brief overview of this approach and its phases.

### 1.6 IBM implementation approach for SAP authorization

IBM follows the phase model on SAP authorization implementation projects (Figure 1-6).

![Figure 1-6 IBM implementation methodology for SAP authorization projects](image)

The IBM implementation methodology for authorization solutions is a proven approach that has been successfully applied at a large number of organizations from various industries. Within the last 15 years this approach has been continuously improved and amended with various templates, tools, and best practices.
The methodology comprises the following phases:

- **Phase 0: Authorization Performance Check**
  - Review of existing authorization concepts, documentation, organizational setup, and governance procedures
  - Review of authorizations in the system to identify current authorization setup and requirements
  - Preparation of result analysis and development of a roadmap

- **Phase 1: Definition of Standards**
  - Development of basic rules for authorizations and authorization governance processes, Authorization Guideline Document
  - Consideration of internal and external requirements (for example, verification of approval process for authorization changes, and so on)
  - Alignment with Audit (internal and external)
  - Definition of naming conventions to ensure a transparent authorization assignment and administration (for example, definition of naming conventions for templates, single and composite authorizations, and so on)

- **Phase 2: Design (Level 1): Definition of Roles on a Functional Level**
  - Definition of job roles and responsibilities for the various client organizational units (for example, small, medium, and large organizations)
  - Preparation and guidance of Role Design Workshops with key stakeholders from the various business departments
  - Definition of functional Segregation of Duties (SoD)

- **Phase 3: Design (Level 2): Detailed Functional-Task-Matrix**
  - Definition of the tasks per job role on activity basis (transactional level)
  - Preparation and guidance of Role Design Workshops with key stakeholders from the various business departments
  - Creation of the Functional-Task-Matrix
  - Definition of transactional SoD
  - Goal: One SoD-conflict-free role per job position for transparency and easy maintenance
Phase 4: Design (Level 3): Organizational-Value-Matrix
- Identification of the required organizational values to realize the required level of detail for the authorization concept
- Creation of the Organizational-Value-Matrix
- Finalization of authorization concept documentation
- Client sign-off

Phase 5: Realization: Creation of Template Roles
- Creation of the template roles based on the Functional-Task-Matrix and authorization standards as defined in phases 1 and 3
- Unit test of template roles

Phase 6: Realization: Creation of Localized Roles
- Creation of the localized roles based on the Organizational-Value-Matrix and authorization standards as defined in phases 1 and 4
- Unit test of localized roles
- Client sign-off

Phase 7: Test, Documentation, and Review
- Preparation and Support of client authorizations tests:
  - Integrations/User Acceptance Test (UAT)
- Creation of a test plan that includes:
  - Positive and negative testing
  - Error tracking and re-test
- Finalization of documentation
- Client sign-off
Phase 8: Define and Finalize Governance Framework

- Verification of the existing governance processes that are of special importance to Audit, such as:
  - User Life Cycle Management
  - Authorization Life Cycle Management
  - Segregation of Duties and Compliance Control Management

- IBM has developed a best practice framework for User Management and Authorization Governance (UMAG) (Figure 1-7) that addresses the following topics:
  - Common Design Principles for Authorization
  - Common Ownership Principles
  - Authorization Change Process
  - Compliance Change Process
  - Business Concepts, which include the requirements towards the Functional and Organizational Model, Ownership and Compliance Requirements
  - User Management Best Practices, which cover audit requirement towards User Change, Revalidation, and Leaver Processes

*Figure 1-7  IBM User Management and Authorization Governance*
Phase 9: Know-how Transfer and Training
- Identifying client training requirements
- Developing training strategy
- Creating training and hand-over material
- Executing trainings

Phase 10: Final Preparation: Setup of User Records
- Collecting user and authorization master data
- Preparing mapping lists for users to roles
- Final transporting of roles to production
- Assigning users to roles
- GO/NO-GO decision

Phase 11: Go-Live Support
- Performing go-live activities
- Providing support in error handling
- Handing over solution to client staff

Phase 12: Monitoring and Review
- Assessing authorization concept
- Reviewing governance processes
- Monitoring SoD status

An SAP authorization project needs a fine-grained plan for successful implementation. The IBM methodology provides a phased approach and is the foundation for further integration activities to extend and enable SAP systems and applications to be part of an enterprise security concept.

More information: Further information about this methodology is outlined in the following IBM publication (available in German language only): SAP Berechtigungswesen - Design und Realisierung von Berechtigungskonzepten für SAP R/3 und SAP Enterprise Portal, Probst / Jäck / et al., SAP Click, ISBN 978-3898423120.

1.7 Conclusion

In this chapter we briefly introduced the business context and general requirements in relation to security and integration with SAP systems and
applications. The IBM Security Framework provides a foundation to meet these requirements and includes a suite of products enabled to support SAP solution specifics. Before we go into more detail about IBM security products and offerings and how they integrate with SAP solutions, we first look at SAP systems and applications. In Chapter 2, “Introduction to SAP solutions and security technology” on page 29, we provide an introduction to SAP solutions and SAP security technology.
In this chapter we discuss the following SAP offerings and concepts for security integration:

- “SAP systems and applications” on page 30
- “SAP security and SAP user and role management concept” on page 34
- “SAP user management integration options and interfaces” on page 41
- “SAP access management integration options” on page 46
2.1 SAP systems and applications

SAP is one of the leading providers of business software. Its product portfolio for enterprise application software is organized around the following key offerings:

- The SAP Business Suite applications are targeted for large organizations and international corporations. The applications support core business operations ranging from supplier relationships to production to warehouse management, sales, and all administrative functions, through to customer relationships. The suite can address specific solutions for 25 industries, such as banking, insurance, chemicals, health care, retail, consumer products, and the public sector.

- SAP Business All-in-One solutions, the SAP Business-By-Design solution, and the SAP Business One application, which address the needs of small businesses and midsize companies.

- The SAP NetWeaver technology platform, which integrates information and business processes across diverse technologies and organizational structures.

- The SAP BusinessObjects portfolio with solutions for business users who need software for analyses, reports, and support to rapidly make strategic decisions, and for relief with administrative tasks. The SAP BusinessObjects portfolio also includes solutions for governance, risk, and compliance (GRC) management to help ensure that customers have the proper processes and controls in place to realize transparent GRC.

Let us look more closely at the SAP Business Suite and the SAP NetWeaver technology platform.

2.1.1 SAP Business Suite

The SAP Business Suite is a family of adaptive business applications that are based on the SAP NetWeaver platform.

From an application point of view, the SAP Business Suite consists of Enterprise Resource Planning (ERP), Central Components (ECC), and Business Suite components:

- SAP ERP
- SAP Customer Relationship Management (CRM)
- SAP Product Lifecycle Management (PLM)
- SAP Supplier Relationship Management (SRM)
- SAP Supply Chain Management (SCM)
This portfolio is complemented by industry-specific solutions that also run as platform-independent. Figure 2-1 shows an overview of the SAP architecture.

![SAP architecture overview](source: SAP AG)

In typical scenarios, a combination of the previously mentioned application components is used. Those landscapes consist of more than one SAP ECC installation, each with its own database. SAP ECC cannot be installed individually. As Java becomes more important and is used for new developments, both ABAP and Java components might be included. In such a case, at least two databases are installed.

To keep these environments manageable, SAP recommends using SAP Solution Manager, a centralized solution management toolset that facilitates technical support for distributed SAP systems, with functionality that covers solution deployment and operation, for example, to keep track of patch levels.
The SAP Enterprise Resource Planning (ERP) product has evolved considerably in the recent past. The original architecture, where the SAP R/3 product ran on the SAP Basis layer, has been changed to one in which ERP is now part of SAP ERP Central Component (ECC) and runs on the SAP NetWeaver Application Server (AS). Figure 2-2 summarizes the changes.

SAP ERP also includes four individual solutions that support key functional areas:

- SAP ERP Financials
- SAP ERP Human Capital Management (HCM, also known as SAP HR)
- SAP ERP Operations
- SAP ERP Corporate Services

### 2.1.2 SAP NetWeaver

SAP NetWeaver is a market branding for a range of software technology, applications, and platform components. The SAP NetWeaver family is built on two foundation technology stacks. These stacks are often referred to as SAP NetWeaver Application Server ABAP (AS ABAP) and SAP NetWeaver Application Server Java (AS Java). As the names suggest, the stacks are essentially application servers or platforms. The Advanced Business Application Programming (ABAP) stack is derived from the SAP R/3 heritage. The Java stack is a fully J2EE-compliant application runtime container. All of the SAP business applications such as SAP ERP, SAP ECC, SAP CRM, and so on, have
components that rely on, and are deployed to, one or both of the SAP AS ABAP and SAP AS Java.

The application server platforms provide many common infrastructure services, including security authentication, authorization, and user account management. Besides the application server, SAP NetWeaver also includes other runtime elements, such as portal and integration layers for processes, information, and people.

Figure 2-3 provides a conceptual illustration of the SAP NetWeaver platform.

---

**SAP NetWeaver Application Server ABAP**

SAP NetWeaver Application Server ABAP provides the development and runtime environment for ABAP-based applications. It also enables selected SAP Java applications to run in the same work process (VM Container).
All ABAP application servers, including the message server, represent the application layer of the multitier architecture of an ABAP-based SAP System. These application servers execute ABAP applications and communicate with the presentation components, the database, and also with each other, using the message server.

In addition to several work processes, whose number and type are determined at the startup of SAP NetWeaver AS ABAP, and that run the actual ABAP program, each ABAP application server contains a dispatcher, a gateway, and the shared memory.

The dispatcher distributes the requests to the work processes. If all the processes are occupied, the requests are stored in the dispatcher queue. The SAP Gateway provides the Remote Function Call (RFC) interface between the SAP instances (within an SAP system and beyond system boundaries).

**SAP NetWeaver Application Server Java**

SAP NetWeaver Application Server Java provides a Java 2 Enterprise Edition (J2EE) 1.5 compliant environment for developing and running Java EE programs.

An SAP AS Java standalone system consists of a database, the central services, and a Java instance. The central services run on one physical machine and are a special type of Java instance. They form the basis of communication and synchronization for the Java cluster. They consist of message and enqueue services. Central services are always required when a Java cluster is installed, and they are identified by system ID and an instance number (for example, C01). A Java instance is a unit in the SAP Web Java cluster, which runs on one physical machine and consists of a Java Dispatcher to dispatch the client request to one of the servers, one or several server processes to run the J2EE application itself, and an instance identified by the system ID.

### 2.2 SAP security and SAP user and role management concept

From the administration perspective, SAP application security is mainly implemented on the technology side with SAP NetWeaver. In this construct, the SAP NetWeaver-based SAP applications can make use of the underlying security functionality. This is especially true for user handling and application authorizations where the concept and use of roles play a vital role.

The administration of this implementation is organized around profiles and authorization objects. Object permissions are the lowest level of access granted
to users providing read, write, and other access. They cannot be assigned
directly to users but can be grouped in any combination to provide the
appropriate access to an application. In this case roles can be created that are
containers for the authorization objects, and at this level the users are assigned
to the specific access granted by the role contents.

For an ABAP application user, a role correlates to a number of transactions that
can execute within an application to again use a certain program.

Authorizations in Java are enforced in the User Management Engine (UME)
using permissions, actions, and roles, whereas an action is a collection of
permissions that can be grouped together into roles. Internally in their Java code,
applications define UME permissions and use them for access control.

Figure 2-4 illustrates the relation of user-role-permission.

To perform an operation in an SAP system, a user might need several
authorizations. In general, users can log on to the system if they have a user
master record with a password. A user menu and authorizations are also
assigned to the user master record through one or more roles. To create and
assign authorizations in ABAP, the Profile Generator is used to automatically
generate and assign authorization profiles (transaction PFCG). The administrator
can also create authorization profiles manually. The assignment of the roles and
profile to a user record then happens through transaction SU01 for dedicated
SAP systems and through the Central User Administration (CUA) for multiple
systems. For Java systems, the AS Java Visual Administrator\(^1\) or the UME configuration and identity management tool can be used to manage J2EE security roles or to manage the user rights for resources of the AS Java.

The tools in Table 2-1 are available from SAP to perform native local user administration functions for SAP systems and applications.

**Table 2-1 SAP user administration tools**

<table>
<thead>
<tr>
<th>SAP user administration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction SU01</td>
<td>User maintenance</td>
</tr>
<tr>
<td>Transaction SU02</td>
<td>Maintain Authorization Profiles</td>
</tr>
<tr>
<td>Transaction SU10</td>
<td>Mass changes in user maintenance</td>
</tr>
<tr>
<td>Transaction PFCG</td>
<td>Role and authorization maintenance</td>
</tr>
<tr>
<td>Transaction SCUA and SU01</td>
<td>Central User Administration (CUA)</td>
</tr>
<tr>
<td>Transaction SUIM</td>
<td>User Information System</td>
</tr>
<tr>
<td>Visual Administrator</td>
<td>SAP J2EE Engine user management using the Visual Administrator(^a)</td>
</tr>
<tr>
<td>UME Console</td>
<td>User Management Engine (UME) administration console (web-based)</td>
</tr>
</tbody>
</table>

\(^a\) The Visual Administrator is used only on Java Stacks 6.40 and 7.00. It is no longer supported in 7.10 (and later) and is replaced by SAP NetWeaver Administrator (Web Dynpro-based UI).

To administer the user accounts on a certain SAP system, you have to know the system specifications, such as whether it is an ABAP or Java environment, and the system connection details (host name and address, SID), and you must have the authorization or administrative privileges to modify the user master record. To perform local user management, these requirements apply to every SAP system and client. In total, that can be a number of applications including test, training, development, QA, and production environments for every SAP ERP, BW, FI/CO, Portal, and so on.

\(^1\) The Visual Administrator is only used on Java Stacks 6.40 and 7.00. It is no longer supported in 7.10 (and later) and is replaced by SAP NetWeaver Administrator (Web Dynpro-based UI).
Figure 2-5 shows a sample deployment.

Next we look more closely at the following items:
- SAP NetWeaver AS ABAP User Repository
- SAP NetWeaver AS Java User Repository: UME
- SAP Central User Administration
- SAP NetWeaver Identity Management
- SAP BusinessObjects governance, risk, and compliance

### 2.2.1 SAP NetWeaver AS ABAP User Repository

For SAP ABAP environments, there is the concept of multiple clients that allows splitting an SAP system into multiple logical sub-systems (clients). This approach is to isolate these sub-systems and operate them as separate business units, for example, to have each customer mapped to exactly one client. All data in a system with multiple clients is located in a common database. An SAP solution can operate with multiple clients if each customer has exclusive access to their data in an installation with a shared system platform, database, and central services. The SAP system itself has separate sets of master data and sets of database tables for each client. Only for client-independent objects and
cross-client tables does a change in one client affect all other clients. All other data is client-specific, such as the user master record. The user administration function itself is an ABAP kernel service and is tightly integrated to the ABAP process administration. The data resides on a database system, consisting of a database management system (DBMS) and the database itself. The ABAP applications do not communicate directly with the database. Instead, they use the kernel administration services, including the user and process services.

In a multi-client ABAP environment with a user who needs access to multiple clients, that user must be configured separately, even if that user resides on the same physical system. This can be achieved to run the transactions, such as SU01, as listed previously, for local user administration and for each client and user master record.

Multiple ABAP clients can be grouped as child systems for central administration of the user data through a CUA master system.

The ABAP user repository and the CUA Central System can be configured to synchronize with directory services through LDAP.

### 2.2.2 SAP NetWeaver AS Java User Repository: UME

The SAP User Management Engine (UME) is the user repository and user administration environment for all SAP NetWeaver Application Server Java-based applications. It can be configured to read and write user-related data from and to multiple data sources. The UME itself runs as a service in the J2EE Engine of the AS Java and is set up as the default user store of the J2EE Engine.

Alternatively, the UME can be configured to use another persistence user data source:

- The SAP NetWeaver Application Server ABAP User Repository can be used for the UME user repository in dual-stack implementations (ABAP and Java). Note that as of SAP Business Suite 7, you can no longer install Dual Stack Application Systems.
- The UME can be configured to use a Lightweight Directory Access Protocol (LDAP) directory server as user persistence store.

### 2.2.3 SAP Central User Administration

For SAP NetWeaver Application Server ABAP-based application user management, SAP provides tools in the form of transactions such as SU01 for user master record creation and maintenance. For large and complex environments, especially when spanned in various business units, managing users
and following company policies is difficult because each SAP instance and client has its own user repository (which again is necessary to gain access to specific SAP applications). To avoid duplicated effort, users can either be managed centrally or get synchronized across all instances. To achieve this on SAP ABAP systems, you can use the SAP module Central User Administration (CUA).

These are SAP CUA characteristics:

- It is available with SAP R/3 Release 4.6.
- It should be a separate SAP instance (CUA Central System or CUA Master) from the business system for the sole purpose of user administration for all other ABAP instances (CUA child systems). An existing systems management solution, such as the SAP Solution Manager, can be used.
- It uses a standard SAP ABAP transaction-based user interface.
- CUA is based on the SAP Application Link Enabling (ALE) distribution model to connect SAP CUA master and child systems.
- CUA data is distributed asynchronously between the application systems in an ALE environment.
- It requires that a person have an identical user ID on all managed instances.
- The use of CUA only works with SAP systems. However, the standard capability to synchronize user data with LDAP directories can be used in a CUA environment also.
- CUA only works for transaction SU01-like activities to synchronize them with target SAP systems. There are no additional approval workflows or identity processes defined.

Before implementing CUA, decide about the following issues:

- Whether to enable local user management at the instance level
  You can define, on a per attribute basis, whether the attribute should be maintained centrally, locally, or redistributed if changed locally or initially be set as proposed.
- Whether SAP HCM should be used to drive user administration based on HR roles
  Using CUA and HR for position-based user administration requires that CUA be carefully designed. The organization has to decide whether position-based role assignments should be managed decentralized (preferred) or centrally in the CUA master system.

Creation of a flexible environment, in which users of several instances are managed centrally and local administrators have full control over their instances,
cannot be solved with CUA. These are other reasons for not using CUA for certain SAP systems:

- Requirements for separate user IDs on specific SAP instances
- Local user administration required on certain SAP instances
- Use of different entitlements for user accounts for the same user on multiple SAP systems
- Password synchronization
- Reduced complexity and staying flexible for future changes
- Workflow scenarios for user management involving the employees and the managers for creation or approving requests
- Rule-based role assignments
- Integration with user management with non-SAP systems

For these scenarios, a product such as IBM Tivoli Identity Manager can come into play, enabling you to integrate an existing CUA environment and SAP stand-alone systems.

As of SAP NetWeaver 7.00, the CUA module entered maintenance mode, which means that no further development or product enhancements will occur, but SAP NetWeaver 7.00 is still available as a product and is fully supported. The SAP strategic product to maintain multiple user records is SAP NetWeaver Identity Management, a product that was acquired from MaXware in 2007.

### 2.2.4 SAP NetWeaver Identity Management

With the introduction of the SAP NetWeaver Identity Management product, SAP closed a gap in the product portfolio to provide a tool that allows maintaining user information that spans SAP systems, including both ABAP and Java platforms. The limitations of the CUA, listed previously, are solved also. SAP NetWeaver Identity Management is offered as an additional installation to ABAP and UME local user administration and CUA. It also provides functionality to integrate third-party identity management tools and non-SAP applications. SAP offers SAP NetWeaver Identity Management services and interfaces for partners to implement solutions, enabling the integration of heterogeneous environments.

The basic components of SAP NetWeaver Identity Management are the Identity Center with the Identity Store and the Virtual Directory Server, which provides the interfaces to third-party identity management vendors.
2.2.5 SAP BusinessObjects governance, risk, and compliance

With the acquisition of Virsa Systems in 2006, SAP started to consolidate its products around governance, risk, and compliance (GRC) management. Since 2008 all the related products reside under the umbrella of the BusinessObjects business unit within SAP.

SAP BusinessObjects GRC is an integrated set of applications that can help to document and manage risks and controls across the enterprise. They can help to automate controls and minimize the likelihood and impact of risks.

The SAP GRC portfolio includes tools for operational and financial risk management, access control, process control, global trade services, and environment, health, and safety management. In combination with enterprise identity management solutions, SAP GRC provides capabilities to define and verify policies and rules for the handling and compliance of identity data. As such, the SAP BusinessObjects Access Control application enforces separation of duties (SoD) across applications and prevents improper access to IT systems.

A GRC integration with Tivoli Identity Manager can be found in 4.4, “Tivoli Identity Manager Adapter for SAP Governance, Risk, and Compliance Access Control” on page 103.

2.3 SAP user management integration options and interfaces

There are several methods for connecting or integrating with SAP back-end systems. Each of the application server platforms provides a graphical interface and API for user account management.

For the ABAP stack, transactions SU01, SU10, and PFCG enable management of user accounts and security roles and profiles. Information that is managed by the transaction is stored in the underlying SAP database.

On the Java stack, the User Management Engine (UME) provides the native user management functions for SAP. The UME provides an abstraction layer, enabling managed data to be persisted in one of these:

- A relational database
- An LDAP directory
- An ABAP application server
As a summary, the connection methods listed in Table 2-2 are relevant when in the context of SAP user account management.

Table 2-2  SAP interfaces for user management

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAPI</td>
<td>Business Application Programming Interfaces provide access to SAP Business Objects such as the user master record. BAPIs are stored as RFC-capable function modules. The SAP user object and BAPIs are defined in the Business Object Repository.</td>
</tr>
<tr>
<td>RFC</td>
<td>The Remote Function Calls interface allows for remote calls between SAP systems or an SAP and a non-SAP system.</td>
</tr>
<tr>
<td>J2EE Security API</td>
<td>Package com.sap.security.api is a web service for SAP user management with NetWeaver AS Java and SAP NetWeaver Portal through the User Management Engine.</td>
</tr>
<tr>
<td>BC-USR-LDAP</td>
<td>Provides integration with SAP user repositories through Lightweight Directory Access Protocol (LDAP):</td>
</tr>
<tr>
<td></td>
<td>• Persistence User Store for UME</td>
</tr>
<tr>
<td></td>
<td>• User data synchronization for ABAP user master records, CUA, or SAP HCM (HR)</td>
</tr>
<tr>
<td>SPML</td>
<td>Service Provisioning Markup Language is an XML-based framework for UME-based user account management.</td>
</tr>
<tr>
<td>ALE/IDOC</td>
<td>Application Link Enabling implements BAPIs for distributed scenarios with asynchronous connections. Intermediate Documents are used to exchange data and information bi-directionally between SAP applications and non-SAP systems.</td>
</tr>
<tr>
<td>JCo</td>
<td>SAP Java Connector is an interface layer for Java programs to access SAP BAPIs/RFC connections.</td>
</tr>
<tr>
<td>GRC-AC-IDM</td>
<td>Web services provided by the SAP BusinessObjects Access Control product Compliant User Provisioning (CUP) enables you to integrate for compliance provisioning of user accounts and role assignments into SAP ABAP servers while ensuring compliance with policy rules and auditing requirements.</td>
</tr>
<tr>
<td>SAP NetWeaver Identity Services</td>
<td>Interface of SAP NetWeaver Identity Management Virtual Directory Server provides support for multiple inbound and outbound protocols:</td>
</tr>
<tr>
<td></td>
<td>• LDAP</td>
</tr>
<tr>
<td></td>
<td>• SPML</td>
</tr>
<tr>
<td></td>
<td>• Web services</td>
</tr>
</tbody>
</table>
Let us take a closer look at the following details:

- Business Application Programming Interfaces (BAPI)
- Remote Function Calls (RFC)
- Synchronous versus asynchronous integration

### 2.3.1 Business Application Programming Interfaces (BAPI)

BAPIs are the standard SAP interfaces to exchange data between SAP components, and between SAP and non-SAP components. BAPIs can be used for an asynchronous type of integration and for synchronous integrations.

BAPIs are Remote Function Calls (RFCs) that represent an object-oriented view of SAP business objects stored in the business objects repository (BOR). The BAPI module accesses the corresponding method that applies to the object.

For example, the RFC module BAPI_USER_GET_DETAIL implements the GetDetail() method for the business object USER.

The business object USER contains user data, such as the logon data or address and communication data. It also contains a read-only reference to the AddressOrg business object.

User management BAPIs are available for:

- User inquiry:
  - BAPI_USER_GETLIST (to obtain a list of users)
  - BAPI_USER_GET_DETAIL (to obtain detailed information about a user, a list of role assignments, and a list of profile assignments)

- Create users:
  - BAPI_USER_CREATE1

- Modify users:
  - BAPI_USER_CHANGE

- Delete users:
  - BAPI_USER_DELETE

- Set initial passwords:
  - BAPI_USER_CHANGE with flag Passwordx and the password in the password field
  - BAPI BAPI_USER_CREATE1 will set an initial password
- Lock and unlock users:
  BAPI_USER_LOCK / BAPI_USER_UNLOCK
- Assign roles:
  BAPI_USER_ACTGROUPS_ASSIGN
- Delete role assignment:
  BAPI_USER_ACTGROUPS_DELETE
- Assign profiles:
  BAPI_USER_PROFILES_ASSIGN
- Delete profile assignments:
  BAPI_USER_PROFILES_DELETE

For functionality that is not provided by standard SAP BAPIs, writing custom BAPI code in ABAP is possible. To load that code into an SAP system in a controlled manner, SAP provides the mechanism of transports. A transport is a special formatted package consisting of several files that include required information and the code itself to implement the new features. The transport import is organized through the transport management system (transaction STMS).

### 2.3.2 Remote Function Calls (RFC)

A BAPI represents a business object. An RFC is the actual functional code and the calling interface for ABAP programs. An RFC enables remote clients to execute ABAP functions.

For certain tasks, use RFCs instead of BAPIs. For example, you cannot use BAPI_USER_CHANGE to change a production (and not the initial) password. For this purpose, you can use the function module SUSR_USER_CHANGE_PASSWORD_RFC.

### 2.3.3 Synchronous versus asynchronous integration

The kind of integration to use with an SAP system to modify user account information depends on a number of criteria. Considerations include the programming language of the calling system, the SAP platform, and whether the request is inbound or outbound. The use of a specific interface also depends on whether the data should be exchanged synchronously or asynchronously.

---

2 In the case of a CUA environment, other BAPIs are used, such as BAPI_USER_LOC_ACTGROUPS_READ, BAPI_USER_LOC_ACTGROUPS_ASSIGN, and BAPI_USER_LOC_PROFILES_ASSIGN.
To explain the alternatives, Table 2-3 shows an example of using an SAP HCM/HR-driven identity feed.

Table 2-3  SAP HR feed alternatives

<table>
<thead>
<tr>
<th>Data exchange</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Export</td>
<td>SAP-triggered export of data to an external storage, for example, a database, LDAP, or file.</td>
</tr>
<tr>
<td>RFC/BAPI Access</td>
<td>SAP RFC program that delivers the requested data per request.</td>
</tr>
<tr>
<td>ALE/IDOC</td>
<td>Remote access from an ALE destination to a queued IDOC file that has been created per pre-defined event or scheduled.</td>
</tr>
<tr>
<td>IDOC through RFC request</td>
<td>The data will be delivered through IDOC when requested through RFC program.</td>
</tr>
<tr>
<td>Direct Access</td>
<td>Direct access to SAP database (Business Object Repository).</td>
</tr>
</tbody>
</table>

Compared to the synchronous integration variants, the asynchronous alternatives provide the benefit that the exchange of data can be controlled much better by the SAP system, but make it more difficult to debug a user provisioning workflow from identity management perspective.

The alternatives in rows 1 - 4 in Table 2-3 provide the best separation of tasks and distinction between SAP and external processes. Row 1 provides the best separation and row 3 provides the most control from the SAP point of view. The variant in row 5 provides the best real-time integration and does not require SAP expertise to set up if the integration is developed that way.

In many cases, the incremental approach is requested: Deliver all changes that happened for a certain time period. That approach is difficult to implement because changes in SAP are often scheduled and take effect only at a definite time. For that reason, a daily feed is most feasible when, for example, IBM Tivoli Identity Manager accepts only the modified records.

See 7.1, “SAP HR driven identity feed” on page 144, for an SAP HR identity feed example using Tivoli Directory Integrator and Tivoli Identity Manager.

In the following sections we discuss the SAP integration options and capabilities provided with IBM Tivoli Directory Integrator and IBM Tivoli Identity Manager in more detail.
2.4 SAP access management integration options

In this section we review various technologies to support single sign-on (SSO) in an SAP solution context. These authentication mechanisms on the SAP platform enable the classical access through the SAP GUI or via interfaces supporting various internet protocols like HTTP, SSL, or SOAP to enable web-based and SOA-based scenarios.

2.4.1 SAP logon ticket

The SAP logon ticket is an HTTP cookie whose data is signed digitally. In addition to the usual intended use of cookies for managing session information and for bypassing the stateless character of HTTP, the text files saved temporarily in the browser are also suitable for web-based SSO to several systems in a common domain, as is the case for the SAP logon ticket.

Before the user can get this ticket, she must log on to an SAP system, which is configured for issuing the tickets, using one of the usual authentication procedures such as user name and password, for example, or her personal certificate. The server uses this logon information to generate the SAP logon ticket and sends it back to the browser as a cookie with the HTTP response. Provided that the user did not configure any restrictions in the security settings in the browser for dealing with cookies, the SAP logon ticket is saved under the name MYSAPSSO2 (Figure 2-6 on page 47) in the local cookie cache of the browser up to the maximum duration of its validity. This is usually restricted to eight hours. If the user closes the browser before the due date expires, the ticket is also deleted.
Like any other cookie, the SAP logon ticket is also subject to technical restrictions that were enforced for data privacy and security reasons. In addition to the maximum size of 4 KB, the browser must only send the ticket to the issuing system or to an adjacent system in the same DNS domain. Thus, the SAP logon ticket remains a valid option for intranet-based SSO in a homogeneous SAP landscape, but it is not recommended for intercompany SSO scenarios, authentication in mixed SAP and non-SAP environments, or when federation of user accounts is required. To validate an SAP logon ticket in an external application running on Windows, UNIX, IBM OS/400®, or IBM z/OS®, SAP programming libraries can be downloaded from the SAP Service Marketplace at the following location:

http://service.sap.com

2.4.2 Secure Network Communication

Secure Network Communications (SNC) is used with the Dynamic Information and Action Gateway (DIAG) protocol in SAP GUI as the front-end client or Remote Function Calls (RFCs) to provide SSO to applications that run in an ABAP environment. Along with the SSO feature, SNC also provides protection at the transport layer.

SNC is a security framework that provides a plug-in architecture for external security products based on the standardized Generic Security Service
Application Programming Interface (GSSAPI, RFC 2743). To support this use case, SAP provides the SAP Cryptographic Library. However, the SAP Cryptographic Library cannot be used for user authentication.

For SSO, the deployment of a third-party security product that has been certified for use by the SAP Software Partner program is required. In a Windows environment, where both SAP GUI and the SAP backend ABAP systems are running on Microsoft Windows, SNC can use Integrated Windows Authentication (IWA) with either NTLM or Kerberos for SSO. SAP provides two implementations of the SNC interface based on IWA to enable SSO:

- The gssntlm.dll library using NTLM
- The gsskrb5.dll library using the Windows Kerberos implementation

For SAP systems installed on other platforms such as IBM AIX®, Linux, or IBM AS/400®, SAP does not provide a library for using SNC with Kerberos. However, there are Kerberos libraries available for non-Windows platforms. Many customers use the original Kerberos release from the MIT, which runs on all UNIX and AS/400 platforms.

### 2.4.3 Digital certificates

Digital certificates are used to verify the identity of the owner of an asymmetric key pair and are issued by trusted organizations called Certification Authorities (CAs). A certificate generally contains a name or other unique designation of the owner of the key pair, the name of the CA that issued the certificate, the validity period of the certificate, the public key from the owner's key pair, and a few other attributes defined by the international ITU-T X.509v3 standard. The CA calculates a digital signature for all this information, which it also adds to the certificate, thus confirming from the CA's side that all the available information about the owner's identity has been checked carefully.

A user can single sign-on to an SAP system using a previously assigned X.509 certificate by establishing a Secure Sockets Layer (SSL) connection with bidirectional (that is, mutual) authentication. After successful verification of the identity information in the certificate, the SAP system authenticates the user without asking for any secret credentials. This step requires a trust relationship between the SAP system and the CA that issued the user's certificate. SSO with X.509 certificates is supported on ABAP and Java-based SAP systems from release 6.40 and later.
2.4.4 Security Assertion Markup Language

The Security Assertion Markup Language (SAML) was introduced by its original authors (Netegrity, RSA Security, Open Network, Sun, and others) in 2001 and ranks among the most successful projects of the industry standards setting organization OASIS. SAML provides a standard for cross-domain and cross-platform SSO for Web Applications by passing authentication information (called a SAML Assertion) based on an XML framework defined by the standard. In this context, SAML does not provide any content-based information about what constitutes an identity and its attributes. Instead, it only specifies the syntax and protocol messages that are used to transfer this information securely over the network in a format that can be uniquely interpreted for every user.

Although SAML was initially developed as an HTTP-based protocol for browser-based SSO-to-web applications, it is also widely used for SOA-based SSO with SOAP Web Services. For Web Services (WS), the OASIS WS-Security standard specifies in a specific security token profile how the SAML Assertion is embedded in the WS-Security SOAP Header to authenticate a WS consumer. These two major scenarios are supported with SAP solutions and are described in more detail in the following sections.

SAML protocol

The two main components of an SAML-enabled landscape for Web Application SSO are an identity provider (IdP) and a service provider (SP), sometimes also called a Relying Party. The SP is a system entity that provides a set of web applications with common session management, identity management, and trust management. The IdP manages identity information for principals and provides a central authentication service to trusted service providers.

Starting with Release 6.40, SAP NetWeaver Application Server (AS) Java supports the use of SAML Versions 1.0 and 1.1 with the Java technology stack, which enables the SAP NetWeaver Portal to act as a SAML 1.x SP for both the Java and ABAP-based Web Applications (for example, Business Server Pages or WebDynpro), but still relies on a trusted third-party SAML 1.x identity provider.

SAP NetWeaver Identity Management 7.1 SP5 introduced a fully compliant implementation of a SAML 2.0 IdP. The corresponding SAML 2.0 service provider is supported with SAP NetWeaver AS Java 7.20 and AS ABAP 7.02 Support Package (SP) 3. Interoperability between SAP and IBM Tivoli Federated Identity Manager (TFIM) 6.2 and other vendor implementations has been successfully demonstrated in 2009 at the Liberty Alliance SAML 2.0 Interoperability Testing based on a variety of SAML 2.0 conformance modes.

---

3 You can find a document about the security token profile here: http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SAMLTokenProfile.pdf
**WS-Security SAML token profile**

The OASIS WS-Security SAML Token Profile is supported in SAP NetWeaver AS 7.00 ABAP and Java to provide authentication at the SOAP message level. The SAML Token Profile defines two methods for how the authenticity of the authenticated user (subject) and its identity information in the SAML Assertion (that is, the SAML Token) can be confirmed to the recipient:

- Sender-Vouches
- Holder-of-Key

**Sender-Vouches subject confirmation method**

The SAML Sender-Vouches subject confirmation method enables SSO for Web services by using a SAML Assertion to forward authentication information acquired in an initial logon. This is typically the case (as shown in Figure 2-7) when an intermediary system, for example, the SAP NetWeaver Portal, acts as the Web Service consumer that has a trust relationship with back-end systems exposing business functionality via SOAP-based Web Services. In this scenario, the Portal (intermediary) is responsible for authenticating the user, for example, based on a password or client certificate, using a web browser client. After successful login, the Portal needs to create an SAML assertion with the user's information that it will add to the back-end Web Service request.

To vouch for the integrity of the SAML assertion and the payload of the web service request, both are signed by the intermediary using a digital signature. The intermediary is able to vouch for the SAML assertion and act on behalf of the user because there is an explicit trust relationship between the back-end system.
and the intermediary, which enables the back-end system to verify the digital signature. After successful verification, the user is logged on to the back-end system and the request is processed. SAML Sender-Vouches-based subject confirmation is the preferred option for system-to-system communications and when the user has a client that is not able to create SOAP messages (such as a web browser).

**Holder-of-Key subject confirmation method**

For passing identity information directly from a desktop or mobile application able to act as a full SOAP WS consumer (like an office application) to a backend system providing a web service, the SAML Holder-of-Key subject confirmation method is better suited. Here, a central authentication and token issuing authority called a Security Token Service (STS) can provision each client with its own key for signing the assertions.
As illustrated in Figure 2-8, the WS consumer embedded in the application first authenticates at the STS ① and requests a new SAML Token. Upon successful authentication of the user, the STS creates ② and issues the security token to the WS consumer ③ who uses it to authenticate the service call to the WS provider ④. The WS provider only needs to trust the STS to verify the SAML Token successfully and authenticate the user. Compared to the SAML Sender-Vouches confirmation method, Holder-of-Key does not require a direct trust between the consumer and provider systems.

![Figure 2-8  Holder-of-Key scenario (source: SAP AG)](image)

The SAML Sender-Vouches confirmation method is supported in SAP NetWeaver AS Java and ABAP from Release 7.00 SP 14. The SAML Holder-of-Key subject confirmation method is supported in AS ABAP 7.01 and later. SAP NetWeaver Identity Management 7.20 supports an STS based on the OASIS WS-Trust 1.3 standard for issuing SAML and X.509 security tokens to SAP and non-SAP Web Service consumers.
2.4.5 Kerberos

Kerberos is an authentication protocol developed by the Massachusetts Institute of Technology (MIT). The authentication process involves several systems connected in a network, or a Kerberos realm. Kerberos authentication within a realm works on the basis of tickets, which serve to prove the authenticity of client requests. Kerberos-enabled authentication with SAP applications makes use of a trusted third-party system, called Key Distribution Center (KDC), which maintains a database of secret keys, where each member system of a realm (whether a client or a server) shares a secret key known only to itself and to the KDC. Knowledge of this key serves to prove the system's identity. After the client is authenticated by the KDC, the KDC generates a session key for securing the communication between the client and the application server. A widely adopted Kerberos implementation is Microsoft Active Directory, where the Domain Controller acts as the KDC.

SAP NetWeaver AS Java supports Kerberos authentication for web browser-based access with the Simple and Protected GSS API Negotiation Mechanism (SPNego). Starting with Release 6.40 SP15, AS Java provides a JAAS-based Login Module to use SPNego to identify itself as a member of a Kerberos realm, determine a shared authentication mechanism with the web browser, and negotiate its use for establishing a security context for further communication with the client.

Refer to SAP Notes 1396724 and 1457499 for more information about the latest update of the SPNego Login Module solution that provides a number of improvements, for example, AES encryption for better interoperability with Windows 7 clients and Windows Server 2008 R2 domain controllers, support for user names with umlaut, simplified setup, and user mapping.

---

4 SAP Notes can be found through the SAP Service Marketplace at [http://service.sap.com](http://service.sap.com) (access to the SAP Service Marketplace requires a valid SAP user ID).
2.4.6 Single sign-on technologies

In this section we introduce the major enabling technologies for single sign-on in an SAP environment. Table 2-4 provides an overview of the technologies and summarizes the supported frontend clients and SAP releases.

Table 2-4 Supported SSO technologies

<table>
<thead>
<tr>
<th>Supported SSO technologies</th>
<th>Supported front-end clients</th>
<th>Supported SAP release</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP logon ticket</td>
<td>Web browser</td>
<td>SAP NetWeaver AS ABAP/Java 6.20</td>
</tr>
<tr>
<td>SNC</td>
<td>SAP GUI, RFC</td>
<td>SAP NetWeaver AS ABAP (all supported releases)</td>
</tr>
<tr>
<td>Digital Certificates</td>
<td>Web browser, Web Service Consumer</td>
<td>SAP NetWeaver AS ABAP/Java 6.40</td>
</tr>
<tr>
<td>SAML Protocol</td>
<td>Web browser</td>
<td>SAP NetWeaver AS ABAP</td>
</tr>
<tr>
<td>SAML Token Profile</td>
<td>Web Service Consumer</td>
<td>SAP NetWeaver AS ABAP 7.00 SP14/AS Java 7.10</td>
</tr>
<tr>
<td>Kerberos/SPNego</td>
<td>Web browser</td>
<td>SAP NetWeaver AS Java 6.40</td>
</tr>
</tbody>
</table>

2.5 Conclusion

In this chapter we introduced the main SAP products and offerings and provided an overview of the SAP security technology for user and access management and its interfaces and integration options. Now that we know about the basic concepts of SAP security let us look at the IBM Security offerings and how they make use of the SAP technology. In the following chapters we introduce the products that IBM provides for identity management and access management.
Identity management integration

In this part, we describe the IBM Security identity management products and offerings for integration with SAP solutions. This chapter contains information about integration points and capabilities that are offered by IBM Security software for identity management of an SAP environment.
IBM Security identity management offerings

In this chapter, we describe the IBM Security identity management products and offerings for integration with SAP solutions. We describe integration points and capabilities that are offered by IBM Security software for identity management of an SAP environment. IBM Security software offers various components and adapters designed to integrate and align with the primary identity management interfaces exposed by SAP applications.

Tivoli identity management offers integrations with SAP NetWeaver Application Server ABAP, SAP NetWeaver Application Server Java User Management Engine (UME), and SAP BusinessObjects Governance Risk and Compliance (GRC) Access Controls. In addition, we integration with the SAP user repositories using Tivoli Directory Integrator and Tivoli Directory Server.

The following products and their integrations with SAP are introduced in this chapter:

- “IBM Tivoli Identity Manager” on page 58
- “IBM Tivoli Directory Integrator” on page 70
- “IBM Tivoli Directory Server” on page 74
3.1 IBM Tivoli Identity Manager

IBM Tivoli Identity Manager is a policy-based identity and access governance solution that helps automate life cycle management of user roles, identities, and access rights.

Through the use of roles, accounts, and access permissions, Tivoli Identity Manager helps automate the creation, modification, and termination of user privileges throughout the entire user life cycle.

Tivoli Identity Manager delivers these:

- A role hierarchy that streamlines administration, provides visibility of user access, and helps bridge the gap between how business users view their IT resources and the actual IT implementation of user access rights.
- Web self-service for managing business roles, accounts, group membership, and passwords.
- Group management to help simplify and reduce the cost of user administration by offering the ability to add, remove, or change the attributes of a group entity within the Tivoli Identity Manager console.
- An embedded workflow engine for automated submission and approval of user requests and periodic certification of user access rights.
- A robust provisioning engine that adds and removes user access rights based on membership in business roles or requests for user accounts and fine-grained entitlements like shared folders or Web portlets.
- A set of controls that enhances security, including preventative separation of duties and closed-loop reconciliation that detects and corrects changes to native target systems.
- Broad, out-of-the-box support for managing user access rights and passwords on applications and systems, plus a rapid integration toolkit for managing custom applications.
- Flexible reporting for user access rights leveraging automatic synchronization of user data from different repositories.

Tivoli Identity Manager delivers automated audit readiness, with certification of fine-grained access rights, separation of duties, closed-loop reconciliation, and prebuilt reports that offer direct auditor access and map low-level IT entitlements into business-friendly descriptions of what users can do with their access.

Closed loop reconciliation features can automatically detect and repair access policy violations that occur due to erroneous changes made on a managed
resource’s administrative console. You can use access rights reconciliation, recertification, and reporting to take these actions:

- Automatically load and reconcile account data.
- Identify and eliminate dormant and ghost accounts.
- Provide ongoing proof for compliance and auditing.
- Maintain records of changes related to access rights.

3.1.1 IBM Tivoli Identity Manager concept

Tivoli Identity Manager enables centralized management and administration of users within the IT environment of an enterprise. Management and administration functions that are provided by Tivoli Identity Manager include these:

- User account provisioning
- User account password management
- Account request approval workflows
- Account recertification
- User access role and group membership management

A large inventory of adapter components enables Tivoli Identity Manager to manage separate distinct IT applications and resources within heterogeneous environments. Adapters are deployed as separate installable units within the infrastructure.

More information: For more details about Tivoli Identity Manager see the IBM Redbooks publication *Identity Management Design Guide with IBM Tivoli Identity Manager*, SG24-6996.

3.1.2 Tivoli Identity Manager adapter concept

Adapters are the primary integration point that enables Tivoli Identity Manager to communicate with target IT systems and resources. Adapters translate Tivoli Identity Manager user account provisioning requests into specific actions on a target IT resource.

Adapters decouple Tivoli Identity Manager from the specific implementation concerns of a target IT resource and vice versa. Adapters typically leverage a remote enabled API of a target resource when executing provisioning requests. Tivoli Identity Manager provides an adapter framework that assists in the development and deployment of adapters.
Figure 3-1 illustrates the generalized adapter architecture.

3.1.3 Adapter operations

Ideally, Tivoli Identity Manager adapters support all of the provisioning operations that might be issued by the Tivoli Identity Manager server, including these operations:

- Add new user account.
- Modify existing user account.
- Delete existing user account.
- Suspend (lock) user account.
- Restore (unlock) user account.
- Change user account password.
- Search for one or more user accounts and return the account details.
- Reconcile supporting attribute data values that can be selected when creating a provisioning request in Tivoli Identity Manager.
- Test connection to both the adapter and the target resource.

The ability of an adapter to support these operations is significantly dependent on the capabilities exposed and supported by a target resource and the target resource API.
3.1.4 Tivoli Identity Manager integration with SAP solutions

Tivoli Identity Manager supports two adapters that enable account management for SAP NetWeaver Application Server Java and SAP NetWeaver Application Server ABAP server stacks:

- **Tivoli Identity Manager Adapter for SAP NetWeaver**
  This adapter enables administration and provisioning of user accounts between Tivoli Identity Manager and SAP NetWeaver ABAP server. It also includes optional integration components, enabling integration between Tivoli Identity Manager and SAP GRC Access Control.

- **Tivoli Identity Manager Adapter for SAP Application Server Java**
  This adapter enables management of user identities in the SAP User Management Engine (UME).

See Chapter 4, “IBM Tivoli Identity Manager” on page 77, for details about the capabilities of these adapters.

3.1.5 SAP user provisioning with IBM Tivoli Identity Manager

In this section we describe why and how to provision SAP NetWeaver from IBM Tivoli Identity Manager.

**Why manage SAP NetWeaver with an identity management solution**

SAP NetWeaver-based systems are often key systems in the corporate environment. They handle critical business transactions and keep track of associated income and expenses. Hence, it is important to ensure that the users with access to the SAP NetWeaver systems have the correct authorizations at the correct point in time. Also, the access to the SAP NetWeaver systems needs to be coordinated with access to other systems, such as Windows Active Directory, Web, and Application Servers upon which the SAP NetWeaver infrastructure can rely.

The SAP license scheme also requires that user licenses are kept in control, to avoid both a missing entitlement and a too-high license cost compared with the actual usage.

Also, the business normally needs to verify compliance and recertification of user access regularly.

These points make the usage of an identity management solution such as IBM Tivoli Identity Manager an obvious choice.
To make the system operational, a provisioning model must be chosen that covers these topics:

- Choice of provisioning model
- Life cycle management of users and accounts
- Management of master data
- Authorization management

The provisioning model

When choosing the provisioning model it is important to understand that the identity management process is an ongoing process. This means that you might start out with a simple model and gradually improve the level of automation and sophistication over time. It is therefore important that the identity management solution allows changes over time and that the implementation project is handled with this in mind.

Figure 3-2 illustrates the choice of provisioning models in the IBM Security Model.

The request-based model is the natural choice for a system with no requirements of tight control (for example, utility systems), but also often is used as a starting point for an identity management solution, as it can be implemented system-by-system without any great effort, and this way quickly get the most critical solution onto the identity management platform.

The role-based model requires a lot of analysis of the corporate business model and requires a considerable effort for defining the role model and its ongoing
management. But it is also the model that makes the compliance and automation of the Identity Management most efficient and coherent.

Due to complexity and the nature of the underlying systems, certain systems will need to be covered by a hybrid model. Also, the time perspective might move a system gradually from a pure request-based system to a role-based system with a hybrid model until the complete role-based model is completed. In the real world the following scenarios are often seen:

- Certain systems (for example, business-critical systems) can by fully RBAC.
- Certain systems (for example, Windows Active Directory) will due to their nature be hybrid.
- Certain systems (for example, utility systems) can be request based.

**User and account life cycle management**

An important part of the identity management solution will be the management of on-boarding, user management, and termination of employees and other users of the corporate IT systems. To ensure that this process is working correctly and in a timely manner it is important that a source of data in good quality is chosen as the data source for the life cycle management. Organizations that rely on SAP solutions often use the SAP HCM (HR) module to manage personnel. Therefore, this is normally a good choice.

The life cycle management process for a person in an organization typically includes a number of steps like those shown in Figure 3-3.

![Figure 3-3 User life cycle management process](image-url)
Account life cycle management will in many situations follow the person’s life cycle, but in many cases accounts will only be created and exist as needed. SAP NetWeaver accounts will normally also follow this general scheme:

- Some might exist throughout the lifetime of a user (for example, for time reporting).
- Some might be linked to a job role (RBAC).
- Some might be project-related.

The provisioning of account must also ensure that credentials are delivered securely to the users and that account password management is covered throughout the lifetime of the accounts. This is especially important if password synchronization is in place. In this case it is important that the IBM Tivoli Identity Manager adapter password synch extension is used to enable password synchronization to be meaningful in the SAP NetWeaver environment.

Another account life cycle aspect to define is the lock/unlock procedures, which must be clearly defined. For example, should a new password unlock an account, and can SAP NetWeaver accounts that have been locked by an SAP system administrator be unlocked by IBM Tivoli Identity Manager?

It is also normal to include a grace period before an account is deleted, which is important if the account contains historical data (for example, an account history) that is not recreated through a re-provisioning of the account. This can involve changes to the Delete operational workflow and certain custom life cycle rules to handle the deletion.

Another important process to cover is the handling of SAP NetWeaver technical users, for example, SAP* and DDIC. IBM Tivoli Identity Manager provides functionality to exclude accounts from the provisioning process completely, which might be a good choice for the SAP* account. For accounts that need to be assigned to users at special occasions or for privileged access requirements, consider a Privileged Identity Management solution solution.

Management of master data

Most IT systems rely on master data (for example, user name), organizational data, and regional preferences that can be derived from the User Master Data used as basis for the user registration in IBM Tivoli Identity Manager coming from a source like the SAP ERP Human Capital Management system. This data is maintained automatically in most cases, but also things such as account default and license data should be handled in the account provisioning process.

There are many ways to achieve this, but with the hierarchical role model in IBM Tivoli Identity Manager, this can be handled by making all authorization roles

---

1 To learn more about the IBM Privileged Identity Management solution see the IBM Redpaper *Centrally Managing and Auditing Privileged User Identities by Using the IBM Integration Services for Privileged Identity Management*, REDP-4660.
depend on a basis role and then linking these to a set of relevant provisioning policies. Figure 3-4 shows how this can be done for two authorization roles and one basis role.

![Diagram showing authorization roles and basis roles](image)

*Figure 3-4  Authorization roles depending on a basis role example*

The linkage to the provisioning policies in this example will then look that illustrated in Figure 3-5.

![Diagram showing provisioning policy and authorization role dependency](image)

*Figure 3-5  Provisioning policy and authorization role dependency example*
The content of the SAP NetWeaver basis policy can be described using Table 3-1, where the values are entered in pseudo code format. In the real policies the JavaScript code must include null checking and relevant null values.

**Table 3-1  Tivoli Identity Manager Provisioning Policy for SAP NetWeaver**

<table>
<thead>
<tr>
<th>Group</th>
<th>Field</th>
<th>Value</th>
<th>Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>User ID</td>
<td>person:preferredUserID</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>First name</td>
<td>person:firstname</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Last name</td>
<td>person:lastname</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Personnel No. (HR only)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Acad. title</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Country</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Function</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Department name</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Room Number</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Floor</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Building</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Communication Language</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Telephone Number</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Telephone Ext.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Fax Number</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Fax Ext</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Email Address</td>
<td>person:mail (in SAP format)</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Communication Method</td>
<td>Email</td>
<td>Default</td>
</tr>
<tr>
<td>Logon data</td>
<td>Internet User Alias</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Group</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Valid From</td>
<td>Current Date</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Valid To</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Account</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Group</td>
<td>Field</td>
<td>Value</td>
<td>Enforcement</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------</td>
<td>--------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td>Cost Center</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>User Type</td>
<td>&quot;Dialog&quot;</td>
<td>Default</td>
</tr>
<tr>
<td>Default</td>
<td>Deactivate Password?</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Start Menu</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Logon Language</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Output Device</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Print Immediately</td>
<td>Checked</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Delete After Print</td>
<td>Checked</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Personal Time Zone</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>CATT Status</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Decimal Notation</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Date Format</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Parameters</td>
<td>Parameter - value</td>
<td>Managed by PP's</td>
<td>-</td>
</tr>
<tr>
<td>Roles</td>
<td>Role Name, Start Date, End Date</td>
<td>Managed by PP's</td>
<td>-</td>
</tr>
<tr>
<td>Profiles</td>
<td>Authorization Profiles</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Groups</td>
<td>User Groups</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SNC Data</td>
<td>SNC Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
  "p:" + toLower(Person:PreferredUserID) + Switch (CompanyID) IT: @IT.MYCOMPANY.COM ADMIN: @ADMIN.MYCOMPANY.COM Other: @MYCOMPANY.NET
  | Mandatory
|               | Insecure Communication Permitted?         | Checked                              | Default              |
| Licence Data  | Contractual User Type                     | 54                                   | Mandatory            |
|               | Country Surcharge (+999 to -100)          | -                                    | -                    |
|               | Assignment To Special Version             | -                                    | -                    |
|               | Substitute Date From                      | -                                    | -                    |
The SNC Data example in Table 3-1 on page 66 shows how it is possible to allow for different values of these data based on the company at which a person is employed.

**Authorization management**

Authorization management covers the provisioning of user rights for the SAP NetWeaver system, which includes Authorization Roles, Authorization Profiles, and Groups.

When managing these objects with IBM Tivoli Identity Manager it is important to understand that Tivoli Identity Manager has no built-in mechanism to avoid conflicting authorization in SAP environments across all objects. It is therefore important that the authorization provisioning model ensures that this does not happen. To reduce the complexity of the model it is a good idea to limit the provisioning of authorization to SAP Authorization Roles only (no profiles/groups allowed for non-technical users).

SAP Authorization Roles have start/end dates as part of their assignment to a user, allowing roles to have a lifespan. This concept might conflict with a role-based model, as business roles might cross system boundaries. Also, the management of dates within the provisioning model in Tivoli Identity Manager is very complicated. We therefore recommend using fixed dates in the provision policies for SAP Authorization Roles, for example, the policy creation date as the start date and 99991231 as the end date. If time-based constraints on roles are needed, we recommend that this is implemented in Tivoli Identity Manager so that the Tivoli Identity Manager roles are being time constrained. This ensures that a role-based implementation is consistent across systems.

In the Tivoli Identity Manager provisioning model, the standard behavior is to allow all values of an attribute if no policy is covering an attribute. Therefore, it is important in the case where a user is covered only by a basis policy that authorizations are disallowed. This is done by adding the value NULL to the basis policy for the relevant attributes. This ensures that, depending on the correction policy, non-allowed authorizations are removed/flagged.

<table>
<thead>
<tr>
<th>Group</th>
<th>Field</th>
<th>Value</th>
<th>Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Substitute Date Until</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Chargeable User SAP System</td>
<td>NULL</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Chargeable User Client</td>
<td>NULL</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Chargeable User Name</td>
<td>NULL</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Receiving System for CUA</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The SNC Data example in Table 3-1 on page 66 shows how it is possible to allow for different values of these data based on the company at which a person is employed.

**Authorization management**

Authorization management covers the provisioning of user rights for the SAP NetWeaver system, which includes Authorization Roles, Authorization Profiles, and Groups.

When managing these objects with IBM Tivoli Identity Manager it is important to understand that Tivoli Identity Manager has no built-in mechanism to avoid conflicting authorization in SAP environments across all objects. It is therefore important that the authorization provisioning model ensures that this does not happen. To reduce the complexity of the model it is a good idea to limit the provisioning of authorization to SAP Authorization Roles only (no profiles/groups allowed for non-technical users).

SAP Authorization Roles have start/end dates as part of their assignment to a user, allowing roles to have a lifespan. This concept might conflict with a role-based model, as business roles might cross system boundaries. Also, the management of dates within the provisioning model in Tivoli Identity Manager is very complicated. We therefore recommend using fixed dates in the provision policies for SAP Authorization Roles, for example, the policy creation date as the start date and 99991231 as the end date. If time-based constraints on roles are needed, we recommend that this is implemented in Tivoli Identity Manager so that the Tivoli Identity Manager roles are being time constrained. This ensures that a role-based implementation is consistent across systems.

In the Tivoli Identity Manager provisioning model, the standard behavior is to allow all values of an attribute if no policy is covering an attribute. Therefore, it is important in the case where a user is covered only by a basis policy that authorizations are disallowed. This is done by adding the value NULL to the basis policy for the relevant attributes. This ensures that, depending on the correction policy, non-allowed authorizations are removed/flagged.
Using the former example with hierarchical roles, the following provisioning examples show how a simple provisioning scenario can be handled. These examples show how Tivoli Identity Manager can support various license types associated with various roles automatically using a priority scheme for the License Data attribute (lower priority gives precedence):

- **Example 1**

  The AUTH1 role 1 gives SAP Authorization Role AUTHA. The priority of the provisioning policy is 1000.

  **Table 3-2 Definition for AUTH1 role**

<table>
<thead>
<tr>
<th>Group</th>
<th>Field</th>
<th>Value</th>
<th>Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles</td>
<td>Role Name, Start Date, End Date</td>
<td>AUTHA 19900101 99991231</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

- **Example 2**

  The AUTH2 role 1 gives SAP Authorization Role AUTHB and changes the license to a more expensive license type. Priority of the provisioning policy is 100.

  **Table 3-3 Definition for AUTH2 role**

<table>
<thead>
<tr>
<th>Group</th>
<th>Field</th>
<th>Value</th>
<th>Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles</td>
<td>Role Name, Start Date, End Date</td>
<td>AUTHB 19900101 99991231</td>
<td>Mandatory</td>
</tr>
<tr>
<td>License Data</td>
<td>Contractual User Type</td>
<td>52</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

In case a user is removed from the role AUTH2, the license type will be governed by the basis policy and hence return the default value.

The SAP system in itself has an Authorization Role hierarchical model called Composite Roles. The IBM Tivoli Identity Manager Adapter for SAP NetWeaver hides this complexity from Tivoli Identity Manager. That is, only the composite roles are reconciled to Tivoli Identity Manager. This is important, as in an automated role-based model it conflicts with the provisioning model. The SAP Authorization Roles derived from the composite role would be disallowed in Tivoli Identity Manager and there would be an attempt to remove them. If necessary, the flexible implementation of the Tivoli Identity Manager Adapter for SAP NetWeaver allows the functionality to be extended so that derived SAP Authorization Roles can be made available in a custom attribute and used for reporting and separation of duty policies.
Managing SAP CUA environment with Tivoli Identity Manager

Tivoli Identity Manager supports provisioning to SAP Central User Administration environments. As this adds another level of complexity on top of the already complex provisioning of SAP NetWeaver, consider this carefully. For SAP NetWeaver systems that are handled through Tivoli Identity Manager as the standard method, we do not recommend having these systems included in the CUA environment, as this does not provide any value to the provisioning process.

Also for systems with tight control (that is, typical production clients), the CUA option is not desirable due to the added complexity.

For lesser control clients (that is, typical test and development clients), the management through Tivoli Identity Manager and CUA might be a desirable option, as some management processes might be simpler, especially if the clients are managed using a request-based provisioning model. In such an environment the Tivoli Identity Manager SAP NetWeaver password extension can be useful also, as it allows propagation of password changes to CUA clients.

Calling custom RFCs in provisioning requests

The IBM Tivoli Identity Manager Adapter for SAP NetWeaver can be extended to call custom RFCs in all provisioning requests. This allows the provisioning processes to extend to support features that are not supported out-of-the-box, for example, activation of Structural Authorizations on creation of a new SAP account or transferring incomplete SAP workflows to another user when suspending an SAP account. See “Extending Tivoli Identity Manager Adapter for SAP NetWeaver” on page 89 for a complete description of this extensibility feature.

3.2 IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator can help organizations build an authoritative data infrastructure, enabling consistent data across multiple identity or generic data resources:

- Transforms, moves, and synchronizes generic and identity data residing in heterogeneous directories, databases, files, collaborative systems, and applications, with real-time automated updates to the authoritative data source.

- Helps enhance the security, accuracy, and integrity of generic and user identity data, while facilitating data migration, transformation to other file formats, and synchronization between two or more systems.
These enhancements are included:

- A simplified interface to develop point-to-point integration
- The ability to visually step through data processed by an AssemblyLine
- The ability to provide Tivoli Directory Integrator as a service using the Representational State Transfer interface

Provides an intuitive graphical user interface for development, deployment, and maintenance of synchronization rules, in addition to a scalable, web-based operations monitoring administrative console.

Provides an open synchronization architecture that supports multivendor IT infrastructures with ease of use, ease of deployment, and rapid time to value, while flexibly scaling from small to very large deployments.

Helps accelerate deployment of IBM security management software such as IBM Tivoli Identity Manager, IBM Tivoli Access Manager, IBM Tivoli Federated Identity Manager, and other IBM infrastructure software, including IBM Tivoli Change and Configuration Management Database (CCMDB), IBM Tivoli Service Request Manager®, IBM WebSphere, IBM Lotus® Domino®, and IBM Lotus Connections middleware.

Supports a broad set of platforms, including IBM AIX, IBM System z®, Microsoft Windows, UNIX, and Linux environments.

Tivoli Directory Integrator integrates and synchronizes generic and identity data residing in a variety of system stores, such as files, message queues, web services, directories, databases, collaborative systems, applications used for human resources (HR), customer relationship management (CRM), Enterprise Resource Planning (ERP), and other corporate applications.

By serving as a flexible, synchronization layer between a company's data infrastructure and the application sources of generic and identity data, Tivoli Directory Integrator eliminates the need for a centralized datastore. For those companies that choose to deploy an enterprise directory solution, Tivoli Directory Integrator can help ease the process by connecting to the identity data from the various repositories throughout the organization. Ensuring that your data is consistent across the organization helps improve the integrity of the information.
With built-in connectors, an open-architecture Java development environment to extend these connectors, and tools to apply logic to data as data is processed, Tivoli Directory Integrator can help you in these ways:

- By synchronizing and exchanging information between files, applications, or directory sources
- By managing data across a variety of repositories, providing the consistent directory infrastructure needed for a wide variety of applications, including security and provisioning
- By creating the authoritative data spaces needed to expose only trustworthy data to advanced software applications such as web services

Tivoli Directory Integrator software architecture includes these:

- An AssemblyLine methodology that builds a compound information object from connected information sources, performs modifications on received data, or creates new entries altogether and adds/updates/deletes the new information object to the assigned destinations.
- An Event Handling Connector framework that adds to the flexibility of Tivoli Directory Integrator by providing the ability to wait for, and react to, specific events that have taken place in the infrastructure, such as changes in a directory, arriving emails, records updated in certain databases, incoming HTML pages from a web server or browser, arriving web services-based Simple Object Access Protocol (SOAP) messages, and other types of events defined by the user.
- Connectors to support numerous protocols and access mechanisms are included with the product or can be easily created or modified.
- Parsers to interpret and translate information from a byte stream into a structured information object, where each piece of information is accessible by name. You can also translate a structured information object into a byte stream. You can select from a wide range of extensible parsers, such as comma-separated values, fixed column, LDAP Data Interchange Format (LDIF), Extensible Markup Language (XML), SOAP, and Directory Services Markup Language (DSML), or create a new parser from scratch.
- Hooks to enable the definition of certain actions to be executed under specific circumstances, or at desired points in the execution of the AssemblyLine process.

The plug-and-play functionality of these components facilitates rapid prototyping and implementation of intelligent data flows. Additionally, it is possible to extend virtually all of these integration components (for example, function components, connectors, and parsers), functions, and attributes through a JavaScript scripting language.
3.2.1 Tivoli Directory Integrator adapter framework

The preferred framework for new Tivoli Identity Manager adapter development is based on Tivoli Directory Integrator and the RMI Dispatcher. Communication between Tivoli Identity Manager and an adapter is based on Java Remote Method Invocation (RMI). Adapter implementations are embodied by Tivoli Directory Integrator AssemblyLines. AssemblyLines ideally use one or more Tivoli Directory Integrator connectors or function components to facilitate target resource interfacing with additional Java or JavaScript processing components.

Conventionally, each adapter operation is delegated to a specific AssemblyLine that specializes in a given operation type. The dispatcher loads Tivoli Directory Integrator AssemblyLine configurations, transferred from Tivoli Identity Manager server to the dispatcher. AssemblyLine configurations are represented in an XML format. The possibility exists for a single dispatcher instance to host multiple Tivoli Identity Manager adapters instances at run time. The RMI provider is the peer component of the dispatcher running in the Tivoli Identity Manager server. Figure 3-6 illustrates the RMI adapter framework.

![Figure 3-6 The architecture of the RMI adapter framework](image)

3.2.2 Tivoli Directory Integrator integrations with SAP solutions

Tivoli Directory Integrator comes with a set of connectors for integration with SAP systems. These connectors are combined within the Tivoli Directory
Integrator Component Suite for SAP NetWeaver Application Server ABAP. Installing IBM Tivoli Directory Integrator also installs the Component Suite. However, to complete the install of the Component Suite, an additional component must be added on the target machine if it does not exist already, which is the SAP Java Connector.

The Tivoli Directory Integrator Component Suite for SAP NetWeaver Application Server ABAP includes these:

- Function Component for SAP NetWeaver Application Server ABAP
- User Registry Connector for SAP NetWeaver Application Server ABAP
- Human Resources/Business Object Repository Connector for SAP NetWeaver Application Server ABAP
- ALE Intermediate Document (IDOC) Connector for SAP NetWeaver Application Server ABAP and SAP ERP

See Chapter 5, “IBM Tivoli Directory Integrator” on page 113, for details about the capabilities of these connectors and the Tivoli Directory Integrator Component Suite for SAP solutions.

### 3.3 IBM Tivoli Directory Server

IBM Tivoli Directory Server software provides a reliable platform for your enterprise security initiatives. This enterprise identity management software from IBM uses Lightweight Directory Access Protocol (LDAP) to provide a trusted identity data infrastructure for authentication:

- Provides identity management for companies that want to deploy a robust and scalable identity infrastructure
- Uses LDAP identity infrastructure software and meets LDAP v3 industry compliance standards
- Enhances proxy server capabilities with flow control for managing requests and paging search results for single and multiple partitions and a smart fail-back mechanism to restore servers safely
- Maintains high availability with master/subordinate and peer-to-peer replication capabilities and scheduled online or offline backup and remote restore
- Supports virtual list views so that you can scroll forward or backward through entries in a large sorted data set and can record deleted entries

2 The SAP Java Connector is available for download for SAP customers at the SAP Service Marketplace at [http://service.sap.com/connectors](http://service.sap.com/connectors) (requires a valid user ID).
IBM Tivoli Directory Server is a powerful and authoritative enterprise directory infrastructure that is a critical enabler for enterprise security. With its open-architecture approach, Tivoli Directory Server can fit well into an existing environment and offers the flexibility to add new directory-enabled IBM or non-IBM applications in the future.

3.3.1 Identity data foundation for security management applications

Tivoli Directory Server provides the identity data foundation for IBM Tivoli integrated identity management and plays a key role in building the enterprise identity data infrastructure for Tivoli identity management services. The strong scalability and flexibility offered by Tivoli Directory Server can benefit third-party applications and IBM solutions for which Tivoli Directory Server is the default directory infrastructure, such as IBM WebSphere Application Server (WAS), IBM WebSphere Portal, IBM Tivoli Identity Manager, IBM Tivoli Access Manager, and the AIX operating system.

Tivoli Directory Server plays a critical role in creating trusted systems that help firms deploy a security infrastructure and embrace web services that open their internal processes to suppliers and customers. Optimized value is brought to companies of all sizes in both the private and public sectors that need a highly scalable, highly available identity data infrastructure for global e-business applications, such as consumer-driven web services.

With IBM DB2® technology as its back-end store, Tivoli Directory Server has exceptional reliability and scalability. Tivoli Directory Server offers robust replication features, including the ability to configure multiple master copies to provide highly available implementations for global enterprise 24x7 support for important business applications.

3.3.2 Tivoli Directory Server integration with SAP solutions

IBM Tivoli Directory Server is an SAP-certified product for integration with SAP using the SAP standard interface BC-LDAP-USR. This integration includes interoperability with SAP NetWeaver Application Server ABAP and Java. As a
result, Tivoli Directory Server can be used as the identity store for synchronization with the AS-ABAP user repository and the persistence user store (user data source) for the AS-Java User Management Engine.

See Chapter 5, “IBM Tivoli Directory Integrator” on page 113, for details about the Tivoli Directory Server and integration capabilities with SAP solutions.

3.4 Conclusion

This concludes the introduction of the IBM Security identity management products and the offerings for integration with SAP systems and applications. We detail each product and its interoperability options with SAP in the following chapters.
IBM Tivoli Identity Manager

In this chapter we describe the Tivoli Identity Manager adapters for SAP solutions. We cover the following topics:

- “Tivoli Identity Manager Adapter for SAP NetWeaver” on page 78
- “Tivoli Identity Manager Adapter for SAP ABAP” on page 101
- “Tivoli Identity Manager Adapter for SAP NetWeaver AS Java” on page 102
- “Tivoli Identity Manager Adapter for SAP Governance, Risk, and Compliance Access Control” on page 103
4.1 Tivoli Identity Manager Adapter for SAP NetWeaver

The Tivoli Identity Manager Adapter for SAP NetWeaver allows the administration and provisioning of user accounts between Tivoli Identity Manager and SAP NetWeaver ABAP-based applications. This adapter is used to provision user accounts and access to SAP business application modules that are deployed on, or use the security infrastructure services of, the SAP ABAP server. The adapter has gained formal SAP integration certification. The adapter supports the following Tivoli Identity Manager provisioning operations:

- User account creation
- User account modification
- User account deletion
- User account suspension (lock)
- User account restoration (unlock)
- Retrieval of user account details
- Password management
- Retrieval of user supporting data, for example, role names that can be assigned to a user
- Linking and retrieval of HR Infotype 0105 (Communication) subtypes to SAP HR personnel records
  
  This is an optional feature and requires SAP HR modules to be installed on a system of the SAP environment.

The adapter enables administration of a significant subset of the user account attributes that can be natively managed using the SAP ABAP User Administration transaction SU01. The manageable attributes include these:

- Basic user details, first and last name, country, department name, and room number and floor
- Communication details, the language that is used to communicate, telephone number, fax number, and email addresses
- SNC name
- Logon data, account validity, and user type
- Account defaults, logon language, output device, decimal notation, and date format
- CUA logical systems associated with the account
4.1.1 Architecture overview

The adapter supports two distinct SAP architectural deployments. The adapter can be deployed against, and administer users directly to, a standalone SAP ABAP server. The standalone system manages its own user, role, and profile registry. The adapter communicates directly with this standalone system. Figure 4-1 illustrates the Tivoli Identity Manager adapter for the SAP ABAP standalone environment.

Alternatively, the adapter can be deployed against a Central User Administration (CUA) SAP architecture. The adapter dynamically determines whether the environment is CUA enabled. The adapter must be configured against the central CUA master server if enabled.

A CUA architecture manages all users from a central master SAP NetWeaver Application Server ABAP system. This master system is assigned child systems. Generally, the master system forwards user administration request actions to the child systems, for example, create user X on child system A. Also possible is for certain actions to flow in the reverse direction from a child system to the master system. This possibility ensures that user account information is synchronized at the central master systems and the child systems within the CUA environment.
Figure 4-2 provides a general architectural overview of the adapter when configured into a CUA environment.

The adapter relies on BAPI and Remote Function Call (RFC) to communicate to an SAP ABAP server and does not need to be deployed on a local SAP ABAP server.

4.1.2 Adapter implementation, customization, and extension

In this section, we describe the implementation details of the Tivoli Identity Manager Adapter for SAP NetWeaver and options for customization and extension.

Adapter architecture

The Tivoli Identity Manager Adapter for SAP NetWeaver is built on the RMI adapter framework. It is composed of AssemblyLines, which process each of the provisioning requests sent from Tivoli Identity Manager. The AssemblyLines use custom Tivoli Directory Integrator connectors and function components that are supplied as part of the adapter distribution package. The connectors and functions components are driven by the configuration of multiple style sheets, which are responsible for formatting and handling RFC calls against the target SAP ABAP server.

The SAP Java Connector (JCo) API provides remote connectivity and access to the target SAP ABAP server. The connectors and function components of the
adapter encapsulate the JCo API. The JCo API is a Java-based programming interface that provides standard function calls and remote network communications to SAP ABAP systems. The library is owned and managed by the SAP application, and because of licensing requirements, it must be downloaded from the SAP support website\(^1\) prior to adapter deployment.

The adapter can be deployed without the need to deploy ABAP code modules or customizations on the target ABAP servers. However, there are often scenarios where functionality is required that is not available with a default installation. These optional extension features are included with the adapter:

- **Single sign-on password**
  
  A single sign-on password enables the adapter to set a productive password. By default, the SAP application allows the adapter to set an initial password for a user account. As a result, the user is prompted to change the password at the next logon.

- **HR Linking**
  
  This extension provides linking and retrieval of HR Infotype 0105 (Communication) sub-types between SAP HR personnel records and SAP user accounts. This extension requires SAP HR modules to be installed on a system of the SAP environment.

- **Account lock management**
  
  This extension prevents the adapter from unlocking SAP accounts that have been locked by the local SAP administrator. This extension allows only the adapter to unlock accounts that have been locked because of failed login attempts. It is usually used in CUA deployments to prevent the adapter from unlocking accounts on a CUA child member server, where a local CUA child administrator might have locked the account.

---

\(^1\) When you connect to the SAP support website, you must have an SAP user ID and password in place: [http://service.sap.com/connectors](http://service.sap.com/connectors)
Figure 4-3 illustrates the adapter architecture.

**Connection management**

Connections and communication between the Tivoli Identity Manager Adapter for *SAP NetWeaver* and the SAP applications are managed by a layered Tivoli Directory Integrator connector design. As Figure 4-3 shows, the user connector and the support data connector encapsulate instances of the RFC function component. The function component is the only layer within the architecture that interfaces directly with the JCo API. Communications between the adapter and the SAP applications are based on the SAP RFC protocol. All variations of SAP connection configuration and connection establishment are supported by the Tivoli Directory Integrator connectors for *SAP NetWeaver* and are exposed and supported by the adapter.

The dispatcher is responsible for the life cycle of the Tivoli Directory Integrator connectors. SAP connection parameter information, defined in the SAP NetWeaver service configuration in Tivoli Identity Manager, is transferred between the Tivoli Identity Manager server and the dispatcher as required. The dispatcher initializes the connectors with the parameter details upon connector instantiation. As a consequence, the connectors attempt to establish and maintain a connection to the target SAP ABAP servers. Connection errors are reported to the Tivoli Identity Manager server.
Add operation

Tivoli Identity Manager operations for adding new user account provisioning are handled by the add AssemblyLine of the Tivoli Identity Manager Adapter for SAP NetWeaver. Figure 4-4 illustrates the adapter execution in response to an add request from Tivoli Identity Manager.

Figure 4-4  Tivoli Identity Manager Adapter for SAP NetWeaver add operation

The account attributes associated with the new account are sent from Tivoli Identity Manager. The dispatcher populates an input work entry and invokes the add AssemblyLine. An instance of the Tivoli Identity Manager Adapter for SAP NetWeaver is deployed in add mode. Tivoli Directory Integrator passes the account attributes to the connector.
The following steps are then executed by the connector within the AssemblyLine (program code) when deployed with default settings and no Advanced Mapping configuration defined in the Tivoli Identity Manager service definition:

1. The input entry is converted to an XML representation.

2. The CUA status of the target SAP ABAP server is determined. The SUSR_ZBV_LANDSCAPE_GET RFC is called to make this determination. The result is cached.

3. If CUA is not enabled, the XSL transforms shown in Table 4-1 are applied in sequence to the input account XML data representation. If the XSL transformation result is not empty (that is, one or more input account attributes were matched), the resulting RFC request is executed.

   Table 4-1 Non CUA add XSL and RFC calls

<table>
<thead>
<tr>
<th>XSL file name</th>
<th>Result XML RFC request</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapnw_bapi_user_create.xsl</td>
<td>BAPI_USER_CREATE1</td>
</tr>
<tr>
<td>sapnw_bapi_user_licensedata.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
<tr>
<td>sapnw_bapi_user_actgroups_assign.xsl</td>
<td>BAPI_USER_ACTGROUPS_ASSIGN</td>
</tr>
<tr>
<td>sapnw_bapi_user_profiles_assign.xsl</td>
<td>BAPI_USER_PROFILES_ASSIGN</td>
</tr>
</tbody>
</table>

4. If CUA is enabled, the XSL transforms shown in Table 4-2 are applied in sequence to the input account XML data representation. If the XSL transformation result is not empty (that is, one or more input account attributes were matched), the resulting RFC request is executed.

   Table 4-2 CUA add XSL and RFC calls

<table>
<thead>
<tr>
<th>XSL file name</th>
<th>Result XML RFC request</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapnw_bapi_user_create.xsl</td>
<td>BAPI_USER_CREATE1</td>
</tr>
<tr>
<td>sapnw_bapi_user_system_assign.xsl</td>
<td>BAPI_USER_SYSTEM_ASSIGN</td>
</tr>
<tr>
<td>sapnw_bapi_user_licensedata.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
<tr>
<td>sapnw_bapi_user_locactgroups_assign.xsl</td>
<td>BAPI_USER_LOCACTGROUPS_ASSIGN</td>
</tr>
<tr>
<td>sapnw_bapi_user_locprofiles_assign.xsl</td>
<td>BAPI_USER_LOCPROFILES_ASSIGN</td>
</tr>
</tbody>
</table>

5. The result of each RFC call is cached before executing subsequent RFC calls. If any RFC call reports a warning, the connector will populate a result work entry with a Tivoli Identity Manager warning status. If any RFC call reports an error, the connector populates a result work entry with a Tivoli Identity Manager failure status. All RFC messages are concatenated and returned to Tivoli Identity Manager.


Modify operation

Tivoli Identity Manager operations for modifying existing user account provisioning are handled by the modify AssemblyLine of the Tivoli Identity Manager Adapter for SAP NetWeaver. Figure 4-5 illustrates the adapter execution in response to a modify request from Tivoli Identity Manager.

![Figure 4-5 Tivoli Identity Manager Adapter for SAP NetWeaver modify operation](image)

The account attributes to be modified and the user name are sent from Tivoli Identity Manager. The dispatcher populates an input work entry and invokes the modify AssemblyLine. An instance of the Tivoli Identity Manager Adapter for SAP NetWeaver is deployed in update mode. Tivoli Directory Integrator passes the account attributes to the connector.

The following steps are then executed by the connector within the AssemblyLine (program code) when deployed in default mode without the Advanced Mapping configuration defined in the Tivoli Identity Manager service definition:

1. The input entry is converted to an XML representation.
2. The CUA status of the target SAP ABAP server is determined. The RFC SUSR_ZBV_LANDSCAPE_GET is called to make this determination. The result is cached.
3. Existing user details are looked up. The connector achieves this by executing the XSL transforms and resulting RFC calls as shown in Table 4-3.

Table 4-3  Lookup user details RFC request for connector in update mode

<table>
<thead>
<tr>
<th>XSL file name</th>
<th>Result XML RFC request</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapnw_bapi_user_getdetail_precall.xsl</td>
<td>BAPI_USER_GETDETAIL</td>
</tr>
</tbody>
</table>

4. If the user lookup RFC response is not an error, execute the XSL transforms shown in Table 4-4.

Table 4-4  Lookup user details RFC response for connector in update mode

<table>
<thead>
<tr>
<th>XSL file name</th>
<th>Result XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapnw_bapi_user_getdetail_postcall.xsl</td>
<td>User account XML, or empty if user does not exist</td>
</tr>
</tbody>
</table>

5. If the user does exist and CUA is not enabled, the XSL transforms shown in Table 4-5 are applied in sequence to the input account XML data representation. If the XSL transformation result is not empty (that is, one or more input account attributes were matched), the resulting RFC request is executed.

Table 4-5  Non CUA add XSL and RFC calls

<table>
<thead>
<tr>
<th>XSL file name</th>
<th>Result XML RFC request</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapnw_bapi_user_change.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
<tr>
<td>sapnw_bapi_user/licensedata.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
<tr>
<td>sapnw_bapi_user/actgroups_assign.xsl</td>
<td>BAPI_USER_ACTGROUPS_ASSIGN</td>
</tr>
<tr>
<td>sapnw_bapi_user/profiles_assign.xsl</td>
<td>BAPI_USER_PROFILES_ASSIGN</td>
</tr>
</tbody>
</table>

6. If user does exist and CUA is enabled, the XSL transforms shown in Table 4-6 are applied in sequence to the input account XML data representation. If the XSL transformation result is not empty (that is, one or more input account attributes were matched), the resulting RFC request is executed.

Table 4-6  CUA add XSL and RFC calls

<table>
<thead>
<tr>
<th>XSL file name</th>
<th>Result XML RFC request</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapnw_bapi_user_change.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
<tr>
<td>sapnw_bapi_user/system_assign.xsl</td>
<td>BAPI_USER_SYSTEM_ASSIGN</td>
</tr>
<tr>
<td>sapnw_bapi_user/licensedata.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
<tr>
<td>sapnw_bapi_user/locactgroups_assign.xsl</td>
<td>BAPI_USER_LOCACTGROUPS_ASSIGN</td>
</tr>
</tbody>
</table>
7. The result of each RFC call is cached before executing subsequent RFC calls. If any RFC call reports a warning, the connector populates a result work entry with a Tivoli Identity Manager warning status. If any RFC call reports an error, the connector populates a result work entry with a Tivoli Identity Manager failure status. All RFC messages are concatenated and returned to Tivoli Identity Manager.

**Delete, suspend, resume, and change password operations**

Tivoli Identity Manager delete, suspend, resume, and change password operations are performed in a similar style to the modify operation described previously. Each operation type is handled by a specific AssemblyLine containing an instance of the SAP NetWeaver User Connector in update mode. Each AssemblyLine determines whether the user account exists in the target SAP ABAP system by executing BAPI_USER_GETDETAIL RFC. AssemblyLines are differentiated by the XSL transforms and executed RFC calls that provide the core functionality of the operation implementation. Table 4-7 summarizes these RFC calls.

<table>
<thead>
<tr>
<th>Provisioning operation</th>
<th>XSL file name</th>
<th>Result XML RFC request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>sapnw_bapi_user_delete.xsl</td>
<td>BAPI_USER_DELETE</td>
</tr>
<tr>
<td>Suspend</td>
<td>sapnw_bapi_user_lock.xsl</td>
<td>BAPI_USER_LOCK</td>
</tr>
<tr>
<td>Restore</td>
<td>sapnw_bapi_user_unlock.xsl</td>
<td>BAPI_USER_UNLOCK</td>
</tr>
<tr>
<td>Change Password</td>
<td>sapnw_bapi_user_change.xsl</td>
<td>BAPI_USER_CHANGE</td>
</tr>
</tbody>
</table>
Reconciliation operation

Tivoli Identity Manager search and reconciliation requests are delegated to the search AssemblyLine. Figure 4-6 illustrates the main components of the search AssemblyLine.

The search AssemblyLine contains three connectors:

- **The User Connector in iterator mode** executes when full reconciliation or a complex filtered search request is received. This connector first obtains a list of the user account names from the SAP application. It then iterates through that list one at a time, retrieving the detailed account attribute information from the SAP application. As each account is retrieved, the details are packaged into a Tivoli Directory Integrator result entry and then queued by the dispatcher for batch sending to Tivoli Identity Manager server.

- **The User Connector in lookup mode** executes when a search request with a filtered query for a specific user name is received. The user name is passed to the connector and used as a parameter to RFC BAPI calls, which retrieves the account attribute details. The details are packaged into a Tivoli Directory Integrator result entry and then passed to the dispatcher for sending to the Tivoli Identity Manager server.

- **The Support Data Connector in iterator mode** executes when a full reconciliation or support data only filtered reconciliation request is received. The connector retrieves lists of support data object class instances using various RFC database table queries. The queries are issued by calling the SAP RFC function named RFC_READ_TABLE. As each object class
attribute data is retrieved, the details are packaged into a Tivoli Directory Integrator result entry and then queued by the dispatcher for batch sending to Tivoli Identity Manager server.

**Extending Tivoli Identity Manager Adapter for SAP NetWeaver**

The Tivoli Identity Manager Adapter for SAP NetWeaver is both highly configurable and highly customizable. With simple configuration, the adapter can be extended to support new attributes, change adapter functionality, or invoke arbitrary custom functions.

In previous sections, we presented an overview of the adapter's architecture and use of declarative programming to define adapter operations. In this section we describe the adapter's extendability and provide examples of adapter customizations.

**Adapter customization types**

The adapter currently supports two customization types, which we describe in this section.

**Supporting the new attribute**

Customizing the adapter to be able to manage new account attributes is one type. Depending on the attribute, different customization options are required:

- The new attribute can be managed with the existing BAPI/Custom RFC module in the adapter.
- The new attribute can be managed with the existing BAPI.
- The new attribute cannot be managed with the existing BAPI and requires a custom RFC module.

**Changing adapter functionality**

Customizing the adapter to perform different functions on a current set of adapter attributes is another type. Depending on functionality, different customization options are required:

- Additional processing of the attribute is required before the corresponding RFC is invoked (for example converting a character from uppercase to lowercase).
- A different order of invoking RFC is required.
- New functionality can be managed with a different existing BAPI.
- New functionality cannot be managed with an existing BAPI and requires new custom RFC.
Supporting new attribute customization
The Tivoli Identity Manager Adapter for SAP NetWeaver supports a substantial subset of the standard SAP ABAP user account attributes. There might be instances when business requirements justify the need to manage new or existing standard account attributes that are not included in the subset supported by the adapter. In these instances, the adapter can be customized to support the additional business requirements.

Typically, the steps to add support for a new attribute in the adapter are as follows:
1. Obtain information about the attribute that you want to support.
2. Add the new attribute to the adapter schema.
3. Modify the account form for the new attribute in the Tivoli Identity Manager server.
4. (Optional) Create a custom RFC if a standard BAPI is not available to perform management for the new attribute.
5. Modify the existing or create a new XSL style sheet to reference a new attribute or invoke custom RFC or standard BAPI. If creating a new style sheet, you need to configure the adapter's service form.

In the following sections we provide an example of how to add a Time Format attribute to the adapter. The Time Format attribute is located on the Defaults tab of the account form in the Tivoli Identity Manager server.

Obtaining information about the attribute
Before you start adding customization for the new attribute, find out what is the available SAP API and the structure for managing this new attribute. This example shows you how to find information for the Time Format attribute. Remember that the following steps are specific for this attribute only:
1. Log on to the SAP GUI, execute transaction SU01, and display a user.
2. Locate the attribute on the account form. In this case, click the Default tab and position your cursor on the text field next to the Time Format attribute (for example, 24 Hour Format, which might be: 12:05:10). Perform these steps:
   a. Press F1, and the Performance Assistant window opens.
   b. Click Technical Information.

   In the Technical Information window, the Filed Data section shows you that the attribute is stored in the table USDEFAULTS with a file name TIMEFM in the format XUTIMFM.
3. Get information about the attribute's possible values and structure, as follows:
   a. Double-click the table name **USDEFAULTS**. The Dictionary: Display Structure panel opens.
   b. Double-click the component type **XUTIMEFM**. The Dictionary: Display Data Element panel opens.
   c. Double-click the domain **XUTIEFM**. The Dictionary: Display Domain panel opens.
   d. Click the **Value Range** tab to see the possible values (Figure 4-7).

![Time Format value range (source: SAP AG)](image)
4. Identify the available API that can be used to manage the attribute in the SAP application, as follows. This specific attribute is managed by standard BAPI functions:

a. Execute transaction code SE37, enter function module name BAPI_USER_CREATE1, and then click **Display**.

The function module BAPI_USER_CREATE1 is displayed in Figure 4-8. The attribute Time Format is stored in the table USDEFAULTS. Entries for this table are managed by the BAPIDEFAULT structure.

![Figure 4-8 Import values for BAPI_USER_CREATE1 (source: SAP AG)](image)

b. To verify that the component TIMEFM is part of the DEFAULTS parameter in the BAPI_USER_CREATE1 function, double-click the BAPIDEFAULT structure and check that the TIMEFM component exists in the BAPIDEFAULT structure.

You now have determined that the Time Format attribute is managed during the add account operation using the standard API call BAPI_USER_CREATE1, in the structure BAPIDEFAULT, as a component TIMEFM with these possible values:

- 0 - 24 Hour Format (for example, 12:05:10)
- 1 - 12 Hour Format (for example, 12:05:10 PM)
- 2 - 12 Hour Format (for example, 12:05:10 pm)
- 3 - Hours from 0 to 11 (for example, 00:05:10 PM)
- 4 - Hours from 0 to 11 (for example, 00:05:10 pm)
**Adding the attribute to the service profile**

To add the attribute to the service profile, follow these steps:

1. Extract the SapNWProfile.jar file using the following command:
   ```
   jar -xvf SapNWProfile.jar
   ```
   Two directories are created:
   - The SapNWProfile directory contains the adapter profile.
   - The META-INF directory contains metadata for the jar file.

2. Delete the META-INF directory because it will be re-created by the adapter profile repackaging in step 5 on page 94.

3. Edit the schema.dsml file in the SapNWProfile directory:
   a. Add the attribute definition to the schema.dsml file. Set the last digit of the object-identifier value to be a bigger number to avoid a clash with future adapter releases. The following example uses 1000. Set all customized attributes names to start with ersapnwcust to be compliant with the current adapter naming convention:
      ```
      <!-- ******************************************************** -->
      <!-- ersapnwcusttimefm -->
      <!-- ******************************************************** -->
      <attribute-type single-value = "true" >
      <name>ersapnwcusttimefm</name>
      <description>Time Format</description>
      <object-identifier>
        1.3.6.1.4.1.6054.3.149.2.1000
      </object-identifier>
      <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
      </attribute-type>
      ```
   b. Append the new attribute reference to the account class section, as follows:
      ```
      <!-- ******************************************************** -->
      <!-- ersapnwaccount class -->
      <!-- ******************************************************** -->
      <class superior="top">
      <name>ersapnwaccount</name>
      <description>sap account class</description>
      <object-identifier>
        1.3.6.1.4.1.6054.3.149.1.1
      </object-identifier>
      ```
4. Repackage the files using the following command from a command prompt:

   jar -cvf SapNWProfile.jar SapNWProfile

5. Import the new SapNWProfile.jar file into Tivoli Identity Manager server.

6. Verify in the trace.log file that the adapter profile was successfully installed.

7. Restart the Tivoli Identity Manager application to refresh the new adapter schema.

**Enabling the attribute in the Tivoli Identity Manager GUI**

To enable the attribute in the Tivoli Identity Manager GUI, follow these steps:

1. Log on to Tivoli Identity Manager Server as an administrator.

2. Select **Configure System → Design Form.**

3. Select **Account → SAP NetWeaver Account.**

4. Select the tab where you want to add the new attribute (for example, $ersapnwtabdefault). From the Attribute list, double-click the **ersapnwcusttimefm** attribute.

5. The attribute $ersapnwcusttimefm appears on the Default tab on the design form.

6. Click the **Sersapnwcusttimefm** attribute, and define a Label Name Time Format under Properties.

7. Right-click the **Sersapnwcusttimefm** attribute, select **Change To**, and then select **DropDownBox**. The DropDown box dialog opens.
8. Click **Custom Values.** In the Select Editor dialog, define values as shown in Figure 4-9.

![Select Editor](image)

*Figure 4-9 Enter custom values*

9. Click **OK** and then save the template.

10. Verify that the new Default tab for the account on the Tivoli Identity Manager server now contains the Time Format attribute.

**Modifying XSL transformation files**

To enable management of the attribute added to the adapter profile and the adapter account form, the corresponding XSL style sheet must be modified. In this example, the adapter is using the default XSL steelfist. If you have customized the adapter and you use a separate XSL steelfist for the account provisioning, modify the corresponding one.

Perform the following steps:

1. To support the new attribute for the add operation, edit the `sapnw_bapi_user_create.xsl` file found in the `ITDI_HOME/timsol/xsl` directory:

   a. Add mapping for the attribute inside the `<DEFAULTS></DEFAULTS>` tag:

   ```xml
   <!-- BEGIN Defaults Structure parameter -->
   <xsl:template name="sapDefaults">
   <DEFAULTS>
   ...
   <!-- Time format -->
   <xsl:apply-templates select="./ersapnwcusttimefm" />
   ...
   </DEFAULTS>
   ```
Changing adapter functionality customization

Typically, the steps for changing adapter functionality are:

1. Create a custom RFC if the standard BAPI is not available to perform the required functionality.

2. Modify the existing or create a new XSL style sheet to invoke the custom RFC or standard BAPI. If you have to create a new style sheet, you need to configure the adapter's service form.

   The following sections provide an example of wrapping all standard BAPI RCF requests during an add account operation into one custom RFC request.
Creating a custom RFC

Use standard SAP development tools to create a new Z_BAPI_USER_CREATE1 module. This module is a simple wrapper that invokes all BAPI modules for the add operation:

- BAPI_USER_CREATE1
- BAPI_USER_SYSTEM_ASSIGN
- BAPI_USER_LOCACTGROUPS_ASSIGN
- BAPI_USER_LOCPROFILES_ASSIGN

**Note:** Handling of license data attributes has been omitted for simplicity.

The module arguments represent a superset of all arguments that are required for the BAPIs. Example 4-1 shows sample ABAP code for the new custom module.

Example 4-1  ABAP code for the new custom Z_BAPI_USER_CREATE1 module

FUNCTION Z_BAPI_USER_CREATE1.

*"---------------------------------------------------------------------
-  "**Local interface:
  ** IMPORTING
  ** VALUE(USERNAME) LIKE BAPIBNAME-BAPIBNAME
  ** VALUE(NAME_IN) LIKE BAPIBNAME STRUCTURE BAPIBNAME OPTIONAL
  ** VALUE(LOGONDATA) LIKE BAPILOGOND STRUCTURE BAPILOGOND
  ** VALUE(PASSWORD) LIKE BAPIPWD STRUCTURE BAPIPWD
  ** VALUE(DEFAULTS) LIKE BAPIDEFAUL STRUCTURE BAPIDEFAUL OPTIONAL
  ** VALUE(ADDRESS) LIKE BAPIADDR3 STRUCTURE BAPIADDR3
  ** VALUE(COMPANY) LIKE BAPIUSCOMP STRUCTURE BAPIUSCOMP OPTIONAL
  ** VALUE(SNC) LIKE BAPISNCU STRUCTURE BAPISNCU OPTIONAL
  ** VALUE(REF_USER) LIKE BAPIREFUS STRUCTURE BAPIREFUS OPTIONAL
  ** VALUE(ALIAS) TYPE BAPIALIAS OPTIONAL
  ** VALUE(EX_ADDRESS) LIKE BAPIADDR3-PERS_NO OPTIONAL
  ** VALUE(UCLASS) TYPE BAPIUCCLASS OPTIONAL
  ** VALUE(FORCE_SYSTEM_ASSIGNMENT) TYPE BAPIFLAG-BAPIFLAG OPTIONAL
  ** VALUE(SELF_REGISTER) TYPE BAPIFLAG-BAPIFLAG DEFAULT SPACE
  ** TABLES
  ** PARAMETER STRUCTURE BAPIPARAM OPTIONAL
  ** RETURN STRUCTURE BAPIRET2 OPTIONAL
  ** ADDTEL STRUCTURE BAPIADTEL OPTIONAL
  ** ADDFAX STRUCTURE BAPIADFAX OPTIONAL
  ** ADDTTX STRUCTURE BAPIADTTX OPTIONAL
  ** ADDTLX STRUCTURE BAPIADTLX OPTIONAL
  ** ADDSMTP STRUCTURE BAPIADSMTP OPTIONAL

  Source file:
  /SM590000
  BAPI_USER_CREATE1
  /SM590000
  BAPI_USER_SYSTEM_ASSIGN
  /SM590000
  BAPI_USER_LOCACTGROUPS_ASSIGN
  /SM590000
  BAPI_USER_LOCPROFILES_ASSIGN
Example wrapper function for creating an SAP user account in single RFC network call.

CALL FUNCTION 'BAPI_USER_CREATE1'
EXPORTING
      username  = username
      name_in   = name_in
      logondata = logondata
      password  = password
      defaults  = defaults
      address   = address
      company   = company
      snc       = snc
      ref_user  = ref_user
      alias     = alias
      ex_address = ex_address
      uclass    = uclass
      force_system_assignment = force_system_assignment
      self_register           = self_register

TABLES
      parameter1 = parameter1
      return    = return
      addtel    = addtel
      addfax    = addfax
      addttx    = addttx
      addtlx    = addtlx
addsmtp = addsmtp
addrml = addrml
addx400 = addx400
addrfc = addrfc
addprt = addprt
addssf = addssf
adduri = adduri
addpag = addpag
addcomrem = addcomrem
groups = groups
parameter1 = parameter1
extidhead = extidhead
extidpart = extidpart.

** Error handling removed for example
CALL FUNCTION 'BAPI_USER_SYSTEM_ASSIGN'
   TABLES
       systems = systems
   return = return.
** Error handling removed for example
CALL FUNCTION 'BAPI_USER_LOCACTGROUPS_ASSIGN'
   TABLES
       activitygroups = activitygroups
   return = return.
** Error handling removed for example
CALL FUNCTION 'BAPI_USER_LOCPROFILES_ASSIGN'
   TABLES
       profiles = profiles
   return = return.
** Error handling removed for example
ENDFUNCTION.

Creating a new XSL style sheet
Example 4-2 shows a new _z_bapi_user_create1.xsl_ style sheet for invoking the new custom module. This example XSL has been created by copying the content of the existing adapter sapnw_bapi_user_create.xsl.

Example 4-2 _z_bapi_user_create1.xsl_ style sheet

```xml
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version="1.0" xmlns:xalan="http://xml.apache.org/xslt">

<xsl:output method="xml"
    indent="yes"
    omit-xml-declaration="yes"
```
cdata-section-elements="BAPI_USER_CREATE1 USERNAME PASSWORD
BAPIPWD
USERALIAS ADDRESS TITLE_P TITLE_ACA1 FIRSTNAME
LASTNAME FROM_DAT
...
item RCVSYSTEM SPEC_VERS COUNTRY_SURCHARGE
SUBSTITUTE_FROM
SUBSTITUTE_UNTIL SYSID CLIENT BNAME_CHARGEABLE
SYSTEMS" />
<xsl:template match="User">
  <xsl:if test="./eruid">
    <Z_BAPI_USER_CREATE1>
      <xsl:apply-templates select="./eruid" />
      ...
      <xsl:call-template name="sapAddress" />
      <xsl:apply-templates select="./ersapnwcuasystem" />
      <xsl:apply-templates select="./ersapnwagrname" />
      <xsl:apply-templates select="./ersapnwprofile" />
      ...
      </Z_BAPI_USER_CREATE1>
    </xsl:if>
  </xsl:template>

<!-- BEGIN Username parameter -->
<xsl:template match="eruid">
  <USERNAME><xsl:value-of select="." /></USERNAME>
</xsl:template>
<!-- END Username parameter -->

<!-- BEGIN Address Structure parameter -->
<xsl:template name="sapAddress">
  <ADDRESS>
    <xsl:apply-templates select="./ersapnwtitle" />
    ...
  </ADDRESS>
</xsl:template>
<!-- END Address Structure parameter-->

<!-- BEGIN Address Structure Fields -->
<xsl:template match="ersapnwtitle">
  <TITLE_P><xsl:value-of select="." /></TITLE_P>
</xsl:template>
...
Configuring the adapter service to use a new style sheet
To configure the adapter to use the new style sheet for the Z_BAPI_USER_CREATE1 custom module, complete the following steps:

1. Copy the z_bapi_user_create1.xsl file to:
   ITDI_HOME/timsol/xsl/
2. Restart the Tivoli Identity Manager Adapter service.
3. On the Tivoli Identity Manager server, specify the file name z_bapi_user_create1.xsl under the Add Advanced Mapping tab on the adapter's service form (Figure 4-10).

4.2 Tivoli Identity Manager Adapter for SAP ABAP
The Tivoli Identity Manager Adapter for SAP ABAP is the predecessor to the Tivoli Identity Manager Adapter for SAP NetWeaver. It is currently supported by
Tivoli Identity Manager development. However, no new features or enhancements will be incorporated into this adapter. The adapter is supported to SAP Basis Version 7.00. New SAP Basis versions are not supported by this adapter. The adapter is functionally equivalent to the Tivoli Identity Manager Adapter for SAP NetWeaver. It allows the administration of users from Tivoli Identity Manager to SAP ABAP stacks.

**Important:** This adapter has been superseded by the Tivoli Identity Manager Adapter for SAP NetWeaver. The Tivoli Identity Manager Adapter for SAP ABAP is in maintenance mode and should not be used in any new deployments.

### 4.3 Tivoli Identity Manager Adapter for SAP NetWeaver AS Java

Tivoli Identity Manager Adapter for SAP NetWeaver Application Server (AS) Java and Tivoli Identity Manager Adapter for SAP NetWeaver are complementary components. Tivoli Identity Manager Adapter for SAP NetWeaver AS Java, however, enables the management of user identities in the User Management Engine (UME) on the SAP AS Java server.

The Tivoli Identity Manager Adapter for SAP AS Java is deployed directly onto the SAP NetWeaver AS Java system as a J2EE Web application archive. This promotes a consistent J2EE deployment model and experience but does so at the cost of adapter functionality that is specific to Tivoli Identity Manager and that is normally provided by the Tivoli Identity Manager adapter framework. An additional consequence is that all user provisioning requests are executed directly against the UME itself, using the processing resources of the hosting application server. Figure 4-11 illustrates the architectural overview of the Tivoli Identity Manager adapter for SAP AS Java.

![Diagram](image-url)  
**Figure 4-11**  
Tivoli Identity Manager Adapter for SAP NetWeaver Application Server Java
The adapter supports these Tivoli Identity Manager provisioning operations:

- User account creation
- User account modification
- User account deletion
- User account suspension (lock)
- User account restoration (unlock)
- Retrieval of user account details
- Password management
- Authorization roles and groups assignment

For deployment information and configuration details see the Tivoli Identity Manager adapter documentation at the Tivoli Identity Manager information center. For Tivoli Identity Manager 5.1 see this website:


4.4 Tivoli Identity Manager Adapter for SAP
Governance, Risk, and Compliance Access Control

The traditional focus of identity management and identity management tools, such as Tivoli Identity Manager, has been to address operational optimization of IT functions. A critical aspect of this is full life cycle management of systems, application accounts, and access. Recent industry trends and evolution have expanded this focus to include concerns generically referred to as *governance, risk, and compliance* (GRC).

Arguably, a foundation of any successful GRC program and technology deployment is identity management that meets the following criteria:

- User account provisioning that enables auditing of account actions
- Attestation and approval of system access and role assignments at the IT operational or business level
- Enforcement of polices, at the time or point of account provisioning, that are designed to enable of Separation of Duties (SoD) while preventing the occurrence of other risk violations

IBM and SAP recognize the identity management foundation and have collaboratively developed integration points between IBM Tivoli Identity Manager and SAP GRC Access Control (AC). The result is a Tivoli Identity Manager adapter that enables provisioning of SAP accounts through the SAP GRC AC suite. To create requests in SAP GRC AC, the Tivoli Identity Manager Adapter for SAP NetWeaver can be configured to convert Tivoli Identity Manager
provisioning requests into SAP GRC AC access requests. Two profiles are provided with the adapter to work with ABAP resources and to provide integration with SAP GRC.

**Note:** Up until Version 5.1.5, the Tivoli Identity Manager Adapter for SAP NetWeaver was offered separately to the Tivoli Identity Manager Adapter for SAP GRC Access Control. Since Version 5.1.6 of the Tivoli Identity Manager Adapter for SAP NetWeaver, the profiles for the Tivoli Identity Manager Adapter for SAP NetWeaver and for SAP GRC Access Control have been combined into a single new adapter. As a result, the Tivoli Identity Manager Adapter for SAP NetWeaver now offers two separate profiles.

The first profile contains SAP NetWeaver adapter account and service attributes only. This profile does not enable a connection with SAP GRC AC. The second profile contains an extended set of account and service attributes necessary to enable interaction with both SAP GRC AC and SAP NetWeaver. This enables Tivoli Identity Manager to coordinate the account compliance checking process in SAP GRC AC with the account provisioning process in SAP NetWeaver. This profile effectively enables a single account provisioning request to perform two tasks:

- Submission of an access request to SAP GRC AC
- Submission of an account provisioning request to SAP NetWeaver, depending on the result of the SAP GRC AC request (that is, approval or rejection)
Figure 4-12 shows the relationships between components of the adapter.

Figure 4-12  Tivoli Identity Manager Adapter for SAP NetWeaver/SAP GRC AC components and relationships
The adapter interfaces with SAP GRC AC using web services. A set of workflow extensions in Tivoli Identity Manager allows the adapter to submit access requests to SAP GRC AC using the SAPGRC_AC_IDM_SUBMITREQUEST web service. Integration with additional SAP GRC AC web services through Tivoli Identity Manager workflow extensions is intended in future adapter releases. The SAP GRC AC web service interface exposes the same features and functions as the Create Request page of the SAP GRC AC Compliant User Provisioning (CUP) web user interface (Figure 4-13).

![Compliant User Provisioning Access Request form](source: SAP AG)

A high level of control is obtained over the provisioning process by configuring Tivoli Identity Manager workflow extensions for the SAP GRC AC application. The Tivoli Identity Manager workflow extensions allow add, modify, suspend, restore, and delete requests to be sent to the SAP GRC AC application. SoD compliance checks are then performed in the SAP GRC AC application prior to provisioning the account in the SAP NetWeaver application. Risk analysis and
remediation features of GRC Compliant Provisioning can be exploited to modify the request, submit an approval, or submit a rejection. In Tivoli Identity Manager workflow there are two possible modes to configure each type of request:

- Non-blocking mode
- Blocking mode

In non-blocking mode, the SAP GRC AC application takes control of target system account provisioning. Following submission of an access request to the SAP GRC AC application, Tivoli Identity Manager workflow continues execution and does not wait for the result of the request in the SAP GRC AC application. This mode allocates responsibility to the SAP GRC AC application for provisioning the account in the SAP NetWeaver application. Figure 4-14 illustrates the integration between Tivoli Identity Manager and the SAP GRC AC application when the workflow extensions have been configured in non-blocking mode.

This is the sequence of operations that occur in Figure 4-14:

1. Tivoli Identity Manager workflow activity invokes a submit request (ADD, MODIFY, SUSPEND, RESTORE, or DELETE) workflow extension.
2. The workflow extension submits the request to the SAP GRC AC application for the approval process.
3. The request workflow is initiated in SAP GRC AC. The CUP approval workflow begins.
4. The SAP GRC AC application submits an approved account provisioning request to the SAP NetWeaver ABAP application.
These tasks must be completed to configure the adapter in non-blocking mode:

1. Import the SAPNWGRCPProfile.jar profile in Tivoli Identity Manager.
2. Install the adapter in Tivoli Directory Integrator and meet its prerequisite requirements.
3. Configure the set of non-blocking workflow extensions in Tivoli Identity Manager for each type of operation (for example, add, delete, and so on).
4. Configure SAP GRC AC to provision accounts in SAP NetWeaver.

In *blocking* mode the Tivoli Identity Manager workflow will block (or wait/pause) following submission of a request to SAP GRC AC. The workflow continues to block until the result of the request has been received from the SAP GRC AC application. A dedicated *notification service* deployed in WebSphere is responsible for periodically querying the SAP GRC AC application, relaying results of completed requests to Tivoli Identity Manager, and un-blocking the relevant Tivoli Identity Manager workflows. This enables Tivoli Identity Manager workflow to become the central point of coordination and auditing for account provisioning. Tivoli Identity Manager assumes responsibility for whether an account should be provisioned in SAP NetWeaver depending on conditions such as whether the request was approved or rejected.
Figure 4-15 illustrates the integration between Tivoli Identity Manager and SAP GRC AC when the workflow extensions have been configured in blocking mode.

This is the sequence of operations that occur in Figure 4-15:

1. Tivoli Identity Manager workflow activity invokes a submit request (ADD, MODIFY, SUSPEND, RESTORE, or DELETE) workflow extension.

2. The workflow extension submits the request to SAP GRC AC for the approval process.

3. The request workflow is initiated in SAP GRC AC. The CUP approval workflow begins.

4. The submitted request identifier is returned to Tivoli Identity Manager, indicating submission success or failure.

5. The workflow extension sets the workflow activity completion status to PENDING. The PENDING workflow activity in Tivoli Identity Manager contains the request ID returned by SAP GRC AC and the activity ID of the Tivoli Identity Manager workflow.

6. The SAP GRC AC compliance workflow completes at some indeterminate time. The notification service receives completed requests in CUP by polling SAP GRC AC on a configurable time period. SAP GRC AC returns a list of request IDs with their completion status (for example, APPROVED, REJECT).
7. The notification service logs into Tivoli Identity Manager and correlates all workflow activities in the PENDING state that have a GRC request ID that matches the ID of a completed GRC request. The status of the relevant workflow activity is updated from PENDING to APPROVED_SUCCESS, APPROVED_REJECTED, or FAILED.

8. If the GRC request was approved, an account request (ADD, MODIFY, SUSPEND, RESTORE, or DELETE) is sent to the Tivoli Identity Manager Adapter for SAP NetWeaver.

9. The Tivoli Identity Manager Adapter for SAP NetWeaver submits an account request to SAP NetWeaver ABAP using RFC.

These tasks must be completed to configure the adapter in blocking mode:

1. Import the SAPNWGRCProfile.jar profile in Tivoli Identity Manager.
2. Install the adapter in Tivoli Directory Integrator and meet its prerequisite requirements.
3. Configure the set of blocking workflow extensions in Tivoli Identity Manager for each type of operation (for example, add, delete, and so on).
4. Install and configure the notification service delivered with the adapter in IBM WebSphere.

To sum up: In this section the concepts that the Tivoli Identity Manager Adapter for SAP NetWeaver has incorporated to coordinate compliant user provisioning with SAP GRC AC have been illustrated.

The Tivoli Identity Manager Adapter for SAP NetWeaver enables two modes for Compliant User Provisioning in SAP GRC Access Control, known as blocking and non-blocking mode. Each mode enables Tivoli Identity Manager to integrate with SAP GRC AC while allowing different options to provision accounts in SAP NetWeaver.

Implementation of the adapter and related components illustrated in this section, such as Tivoli Identity Manager workflow configuration, are presented in further detail in the IBM Tivoli Identity Manager Version 5.1 Adapter for SAP NetWeaver Installation and Configuration Guide, SC23-9660, which can be found at the following location:

https://www.ibm.com/support/docview.wss?uid=pub1sc23966000
4.5 Conclusion

This concludes our discussion of the IBM Tivoli Identity Manager product and its adapters provided for the integration with SAP systems and applications. In the next chapter we investigate the IBM Tivoli Directory Integrator product and its integrations with SAP solutions.
IBM Tivoli Directory Integrator

Tivoli Directory Integrator is a lightweight framework that enables simplified access and synchronization of data among disparate data sources and stores. It is primarily intended for identity-related data integration, but is flexible for any type of data. It has a substantial catalogue of connectivity components for various data stores. Included in the catalog is a suite of components and connectors for SAP NetWeaver Application Server ABAP-based integration.

We discuss the Tivoli Directory Integrator component suite for SAP in this chapter. This chapter contains the following sections:

- “Function component for SAP NetWeaver AS ABAP” on page 114
- “User Registry Connector for SAP NetWeaver AS ABAP” on page 117
- “HR/Business Object Repository Connector for SAP NetWeaver AS ABAP” on page 119
- “IDOC Connector for SAP ERP and SAP NetWeaver AS ABAP” on page 121
5.1 Function component for SAP NetWeaver AS ABAP

The SAP RFC function component enables a Tivoli Directory Integrator AssemblyLine to access SAP data by executing SAP ABAP code exposed by a Remote Function Code module (RFC). RFCs are code modules implemented in the SAP ABAP programming language. RFCs can be invoked by external processes and other connected SAP servers. Depending on the requirements and design, an RFC module can read or write data into the SAP database. This point is significant, because SAP does not support access to its database other than through ABAP code modules. SAP does not support processes that directly access the underlying database tables. SAP supports an extensive range of RFC modules for all types of data. The Tivoli Directory Integrator RFC Function Component can invoke any of these standard or custom modules.

Tivoli Directory Integrator Connector documentation: Detailed information about the installation, configuration, usage, and customization of the Tivoli Directory Integrator Component Suite for SAP NetWeaver Application Server ABAP can be found here:

The Tivoli Directory Integrator Function Component for SAP NetWeaver Application Server ABAP uses the SAP Java Connector (JCo) to invoke RFCs on the AS-ABAP System. The function component provides a means of calling an arbitrary RFC. Figure 5-1 illustrates the overview architecture of the RFC Function Component.

![Overview architecture of the RFC Function Component](image)

The RFC Function Component projects an XML representation of the RFC parameter interface into the Tivoli Directory Integrator AssemblyLine. The component dynamically retrieves the RFC interface metadata from SAP applications. The metadata is then used to parse and process the XML. After the RFC module has been invoked and the resulting data returned from SAP applications, the interface metadata is again used to serialize the data into an XML format for AssemblyLine consumption.

To invoke an RFC using the function component, the AssemblyLine must create an XML document consisting of the RFC’s import parameters. Import parameters can be simple or structured data types. The RFC might require the caller to
supply table parameters, which are similar to arrays and might be required as input to the call. Example 5-1 shows a generalized RFC XML request.

Example 5-1  Generalized RFC XML request representation

```xml
<RFC_MODULE_NAME>
  <SIMPLE_IMPORT_PARAMETER>value</SIMPLE_IMPORT_PARAMETER_A>
  <STRUCTURE_IMPORT_PARAMETER>
    <STRUCTURE_FIELD_A>value</STRUCTURE_FIELD_A>
    <STRUCTURE_FIELD_B>value</STRUCTURE_FIELD_B>
  </STRUCTURE_IMPORT_PARAMETER>
  <TABLE_PARAMETER_A>
    <item>
      <FIELD_A>value</FIELD_A>
      <FIELD_B>value</FIELD_B>
    </item>
    <item>
      <FIELD_A>value</FIELD_A>
      <FIELD_B>value</FIELD_B>
    </item>
  </TABLE_PARAMETER_A>
</RFC_MODULE_NAME>
```

The RFC Function Component returns a similar XML format, with the exception that the RFC's export and tables parameters are present (Example 5-2).

Example 5-2  Generalized RFC XML response representation

```xml
<RFC_MODULE_NAME.Response>
  <SIMPLE_EXPORT_PARAMETER>value</SIMPLE_EXPORT_PARAMETER_A>
  <STRUCTURE_EXPORT_PARAMETER>
    <STRUCTURE_FIELD_A>value</STRUCTURE_FIELD_A>
    <STRUCTURE_FIELD_B>value</STRUCTURE_FIELD_B>
  </STRUCTURE_EXPORT_PARAMETER>
  <TABLE_PARAMETER_A>
    <item>
      <FIELD_A>value</FIELD_A>
      <FIELD_B>value</FIELD_B>
    </item>
    <item>
      <FIELD_A>value</FIELD_A>
      <FIELD_B>value</FIELD_B>
    </item>
  </TABLE_PARAMETER_A>
  <TABLE_PARAMETER_B>
    <item>
      <FIELD_A>value</FIELD_A>
      <FIELD_B>value</FIELD_B>
    </item>
  </TABLE_PARAMETER_B>
</RFC_MODULE_NAME>
```
Although Tivoli Directory Integrator is an IBM Tivoli security product, the RFC Function Component can integrate with any security-related or non-security-related data in an SAP application that is exposed through an RFC interface. It can be considered a low-level technology access component. It is suited to scenarios that require maximum control and flexibility of solution design. Use of this component typically requires the user to have some ABAP skill, and certainly enough knowledge and understanding of the RFC interfaces that are to be called.

5.2 User Registry Connector for SAP NetWeaver AS ABAP

The Tivoli Directory Integrator User Registry Connector for SAP NetWeaver Application Server ABAP enables management and provisioning of SAP AS ABAP accounts. It supports the basic create, read, update, and delete (CRUD) operations for ABAP user accounts and can be deployed in any of the following standard Tivoli Directory Integrator connector modes:

- **Add Only**  
  For creating new ABAP user accounts
- **Update**  
  For modifying ABAP user accounts
- **Delete**  
  For removing ABAP user accounts
- **Lookup**  
  For reading individual ABAP user account data
- **Iterator**  
  For reading user account data for all users on an ABAP server
The connector integrates with SAP ABAP servers using the SAP RFC API and network protocol. The connector provisions ABAP user accounts by invoking the standard SAP Business APIs (BAPIs) for identity management. As a result, the connector does not require deployment of custom RFC ABAP code onto the target ABAP server. Figure 5-2 illustrates the component design of the SAP User Registry Connector.

The connector projects an XML representation of ABAP user account data into the Tivoli Directory Integrator AssemblyLine. For each connector mode, one or more XSLT style sheets transform and map parts of the user XML representation into RFC/BAPI calls. The connector uses the lower-level Tivoli Directory Integrator RFC Function Component to execute the actual RFC call against the target ABAP server.

The User Registry Connector has much functional overlap with the Tivoli Identity Manager Adapter for SAP NetWeaver in terms of pure provisioning capability. However, it is first and foremost a Tivoli Directory Integrator connector. It is
intended for users who want to take more responsibility and gain flexible and customized control of provisioning of SAP ABAP accounts.

5.3 HR/Business Object Repository Connector for SAP NetWeaver AS ABAP

The Tivoli Directory Integrator Human Resources/Business Object Repository Connector (HR/BOR Connector) enables Tivoli Directory Integrator AssemblyLines to synchronously access any SAP master data. It does this by leveraging the synchronous Business API (BAPI) interfaces of the Business Object Repository (BOR). Among these interfaces are SAP HR BAPIs.

With such a wide range of possible SAP HR Infotype data attributes and BOR BAPI interfaces, the Tivoli Directory Integrator connector has been designed for flexibility and customization. The flexibility is delivered by strong reliance on XML and XSLT. The connector projects an XML representation of SAP BOR data into the Tivoli Directory Integrator AssemblyLine. The XML data is transformed by XSLT style sheets into BAPI calls, which are executed on the target SAP ABAP server.
Figure 5-3 illustrates the component design of the Human Resources/Business Object Repository Connector for SAP NetWeaver Application Server ABAP.

The default distribution of the Tivoli Directory Integrator SAP HR/BOR Connector ships with and uses XSLT style sheets that enable access and management of SAP HR personnel data defined by Infotype 0002. Infotype 0002 attributes are considered most relevant to identity management scenarios. However, it is acknowledged that other infotypes are also applicable. In these cases, users of the connector are able to define alternate XSLT style sheets that drive the connector to access and manage these data attributes.

The Tivoli Directory Integrator HR/BOR Connector enables read and write access to data. When deployed in iterator mode, the connector can form the basis of an initial HR feed bulk load into Tivoli Identity Manager. With sufficient AssemblyLine logic, the connector can also enable scheduled periodic update of HR identity data in Tivoli Identity Manager.
Depending on the BAPI interfaces available, the connector is able to support the connector modes in the following list (all modes listed are supported for default access to Infotype 0002 attributes):

- **Add Only** For creating business objects
- **Update** For modifying business objects
- **Delete** For removing business objects
- **Lookup** For reading individual business objects
- **Iterator** For reading business object instances of a given type

### 5.4 IDOC Connector for SAP ERP and SAP NetWeaver AS ABAP

Tivoli Directory Integrator supports integration with SAP ABAP servers through the Application Link Enabling (ALE) Intermediate Document (IDOC) interface. This interface supports asynchronous event style access to SAP Master Data exposed through the BOR. ALE and IDOCs are traditional core technologies defined by SAP and are supported by numerous SAP Master Data and application modules. Pushing data updates into an SAP ABAP server in the form of an IDOC data representation is possible. Also possible is for registered listeners to receive IDOC data as the result of a business event or data update originating in the ABAP server. Conceptually, IDOCs are hierarchical data structures representing the state of an SAP business object. The network transmission of IDOCs between two ALE partners is governed by the SAP Transactional RFC protocol. This protocol is layered on the standard RFC protocol and ensures that IDOCs are delivered at least once and are not lost during transmission.

The Tivoli Directory Integrator ALE IDOC Connector is able to receive IDOCs from an SAP ABAP server. It cannot send IDOCs to an SAP application. The connector registers its connection with the SAP gateway service. The connector enables Tivoli Directory Integrator to act as an external RFC server with respect to SAP applications. ALE distribution models defined on the SAP ABAP server describe and govern the relationship between two ALE partners. The Tivoli Directory Integrator ALE IDOC Connector can be configured as the destination or the receiving partner in any ALE distribution model. The connector is able to receive and parse any IDOC message type. When an IDOC is received by the connector, the connector dynamically retrieves the IDOC schema metadata from SAP applications and then converts the IDOC data into an XML representation. Other components in the Tivoli Directory Integrator AssemblyLine are responsible for parsing and processing the IDOC XML as required. The connector supports iterator mode only.
SAP has predefined ALE distribution models for SAP HR data. The HR distribution models allow HR partners to exchange IDOC message types named HRMD_A. IDOCs of this message type are capable of containing many HR Infotypes in a single instance. The traditional scenario for an HR distribution model is to enable the synchronization of employee information between head office and regional offices. In a typical example, organizational structure and planning are performed at the head office. These changes are pushed out to the regional office partners by ALE based on a configured distribution model. Hiring, data maintenance, and termination of employees are performed in the regional offices. These changes are pushed into the head office by ALE. By defining the Tivoli Directory Integrator connector as a receiving partner, the possibility exists for a Tivoli Directory Integrator AssemblyLine to participate in these types of scenarios. From an identity management perspective, participating in HR life cycle scenarios enables integration points that can automate provisioning and de-provisioning of user accounts for employees.

Given the asynchronous model of ALE, this connector is an excellent fit for solutions that require SAP application-driven refresh of HR identity data. The connector can enable an HR feed into Tivoli Identity Manager of updated identity data. See 7.1, “SAP HR driven identity feed” on page 144, for an example.
5.5 Conclusion

This concludes our discussion of the IBM Tivoli Directory Integrator product and its integrations with SAP solutions. In the next chapter we look at IBM Tivoli Directory Server and its interoperability options with SAP solutions.
IBM Tivoli Directory Server

IBM Tivoli Directory Server is a powerful, security-rich, and standards-compliant enterprise directory for corporate intranets and the internet. IBM Tivoli Directory Server is built to serve as the identity data foundation for rapid development and deployment of your web applications and security and identity management initiatives by including strong management, replication, and security features. As an LDAPv3-compliant directory server it is certified and supported by SAP for interoperability with SAP NetWeaver Application Server ABAP and Java.


**LDAP configuration and schema files for use with IBM Tivoli Directory Server and SAP applications:** Here are details about finding required LDAP schema extension and configuration files with appropriate notes for SAP applications

- See SAP Note 1352688 for instructions about how to obtain the IBM schema extension for SAP ABAP and IBM Tivoli Directory Server.
- See SAP Note 1015699 for SAP Java UME settings with IBM Tivoli Directory Server.
6.1 Integration with SAP NetWeaver AS ABAP

SAP NetWeaver Application Server ABAP is able to perform a synchronization of its user master data with IBM Tivoli Directory Server using the LDAP protocol. A mapping configuration allows you to customize the object classes and attribute types that will be used for the directory entries representing users of the AS-ABAP-based applications.

**User synchronization:** This user synchronization only applies between the AS-ABAP user repository and the LDAP directory, and not the persistence user store as with AS-Java and UME.

6.1.1 Definition of attribute mappings

The user master data synchronization of the SAP NetWeaver Application Server ABAP uses a fixed list of data that can be synchronized with Tivoli Directory Server. These attributes must be mapped to attributes in the directory. If fields are identified that cannot be mapped to a default directory attribute, a schema extension will be necessary.

6.1.2 Definition of required schema extension

For those fields of the SAP ABAP user master data that do not fit into the delivered schema of the directory, one must make a schema extension.

The synchronization process will use the SAP-defined object class `sapAddOnUM` that carries the SAP ABAP attributes and will identify an entry in the directory as relevant for synchronization.

The schema extension file for Tivoli Directory Server is provided to perform the directory schema extension for the required SAP attribute types and the object class. See SAP Note 1352688 for instructions about how to obtain the IBM schema extension for SAP ABAP.

6.1.3 SAP NetWeaver AS ABAP LDAP configuration

For synchronization of the SAP NetWeaver Application Server ABAP user store with an LDAP directory server there is a tool that comes with AS-ABAP that needs to be used for that purpose. That tool is the **LDAP Connector** to start with transaction code LDAP.
For synchronization of the SAP NetWeaver Application Server ABAP user store with an LDAP directory server, AS-ABAP provides the LDAP Connector tool, which is started with transaction code LDAP. After you have started the LDAP Connector you need to tie it to the LDAP Client library for your operating system. If the Connector input field already shows a value, you already have a configured LDAP Connector. On some hardware platforms, the LDAP client library is not part of the operating system and must be installed separately. These platforms are listed in a document specified in SAP Note 188371.

For information about LDAP Client Libraries for SAP Web Application Server ABAP see the following link:


Specific information about how to operate the LDAP Connector on Linux and requirements for different versions of LDAP Client Libraries can be found in SAP Note 1380245.

To make the SAP LDAP Connector work and known to the system, an RFC connection needs to be created. In transaction SM59 create an RFC Destination with the following parameters:

- **Name**: LDAP_Connection.
- **Type**: T.
- **Description**: LDAP_Connection.
- **Activation Type**: Registered Server Program.
- **Program ID**: LDAP_Conn.
- **Gateway host**: Enter the hostname of the SAP WebAS.
- **Gateway service**: Enter sapgw<xx> where <xx> is the instance number of the SAP Web Application Server.

Back in the LDAP Connector settings, the RFC details used above are required where the connector name is the RFC destination name.

The LDAP connector is started automatically by CCMS after a maximum time of 10 minutes. It can also be started manually by selecting Start Connector.

To log on to the directory in the background, the AS-ABAP must store the access data. In the transaction LDAP, select System Users, go to change mode, and select New Entries. Create an entry with the following data:

- **User ID**: LDAP_CONN.
- **Distinguished Name**: The DN of the user to access the directory.
Only read auth.  Do *not* select this check box.

Auth. mechanism  Only "Simple Bind" is supported.

Credential storage  "Secure Storage".

If you encounter problems using secure storage and the option is available, use *Simple Memory*. Select **Change** and enter the password twice for access to the directory. To configure the *LDAP Server connection* settings in transaction *LDAP*, select **LDAP Servers**, go to change mode, and select **New Entries**. In this dialog you can enter the connection details for your Tivoli Directory Server. These are the parameters:

- **Server name**: LDAP_CONN (LDAP Connector as specified above).
- **Host name**: The hostname of the directory server.
- **Port Number**: The port to be used (in most cases 389 or 689 when SSL is used).
- **Product name**: Select the entry **Other certified product**. The content of this field is irrelevant for the runtime behavior of the SAP system.
- **Protocol version**: Choose **LDAP version 3**.
- **LDAP Application**: Choose **User**.
- **Default**: Select the check box.
- **Base entry**: Enter the DN of the root entry for all the following operations (directly underneath this entry the SAP Application Server will create the users, and anywhere underneath this, entry users for synchronization will be searched).
- **System Logon**: Enter LDAP_CONN (this entry points to the data entered in previous section of this chapter).

**Read Anonymously**: Do *not* select this check box.

Then double-click **Mapping** in the left tree.

**Note**: Mapping defines only the data conversion and data layout, while synchronization controls the operative behavior.

You can copy an already existing mapping by the menu function, or you can use the XML import function to import a proposal from SAP Note 1352688 for IBM Tivoli Directory Server. In this case, go to **XML Import** of the Mapping Overview screen and select the attribute mapping XML file.
By default, SAP users in the directory will be identified by the object classes inetOrgPerson and sapAddOnUM. As an example for a field mapping, ADDRESS-LASTNAME is mapped to attribute sn.

**Certified LDAP servers:** Not all SAP-certified LDAP servers are displayed in the list field *Certified LDAP directory servers* (LDAP Server Maintenance, *Server view*) and on the selection screen of the RSLDAPSCHEMAEXT report in the SAP NetWeaver Application Server ABAP.

The certification process for the BC-LDAP-USR interface (Directory Interface for User Management via LDAP) was changed regarding the organizational procedure. (No changes were made regarding the technical checks.) The certified products no longer have to be listed in the list field.

See SAP Note 983808 for more information about this field.

### 6.1.4 Directory synchronization

To perform SAP NetWeaver Application Server ABAP user master synchronization with Tivoli Directory Server, the RSLDAPSYNC_USER report will be used for this task. The synchronization tool interprets SAP users below the base DN of the directory server per entries having the object classes configured in LDAPMAP for the LDAP Server and the filter attribute. Use transaction LDAPMAP to modify these settings. The LDAP integration allows AS-ABAP to look at the directory data and change SAP user data outside of SAP applications.

**More information:** For detailed information about how to synchronize the ABAP User Store with an LDAP Directory see this website:


To start the synchronization, execute the RSLDAPSYNC_USER report. In a production environment this report can be started as a background job for delta synchronization (for example, using transaction SA38).

**Exclusions in synchronization:** Certain system users are excluded from the synchronization, such as SAP*, DDIC, and so on. Other communication users, such as those for RFC connections, are treated like normal users.
6.2 Integration with SAP NetWeaver AS Java

With SAP NetWeaver Applications Server Java-based applications and components like SAP NetWeaver Portal you will find the User Management Engine (UME) that handles user registry-related information. UME can be configured to store common user data to an external LDAP directory. In the past SAP delivered pre-configured XML configuration files for several directories. Because the list of directory servers shipped with SAP NetWeaver Portal does not get updated, it does not reflect the current state of supported and certified LDAP directory servers. This is important to know because IBM and other vendors have certified their products for use with UME and SAP NetWeaver Application Server Java, and UME and the SAP NetWeaver Portal. To get the UME data source files for Tivoli Directory Server, see SAP Note 1015699.

6.2.1 General overview of the basic component LDAP interface

The SAP NetWeaver Application Server Java UME LDAP specification for BC-LDAP-USR 6.30 describes the general UME LDAP capabilities and requirements, as stated below.

Consistent user management requires the integration of the numerous data repositories scattered throughout the enterprise. The UME enables you to leverage your existing system infrastructure by accessing user-related data on an existing corporate directory, a database, or an SAP system.

With UME you can connect to an LDAP directory server using an LDAP persistence adapter. You can even read data from and write data to multiple physical LDAP directory servers, or different branches of the same LDAP directory server.

Entries in an LDAP directory server are organized in a tree-like structure called the Directory Information Tree (DIT). UME supports certain methods of arranging users and groups in a DIT in the LDAP directory server:

- Groups as tree (deep hierarchy)
- Flat hierarchy

You can configure secure connections using the Secure Sockets Layer (SSL) protocol between UME and an LDAP directory server to encrypt data transferred between the two parties (client and server).
UME also supports data partitioning. This means that you can use different data sources for different user sets or attribute sets. You can partition data in two ways:

- **User-based data partitioning**
  
  Different sets of users are written to different data sources. For example, in a collaboration scenario, where both users internal to your company and users from other companies work together in the same application, the external users need a user account too. In this case you can configure the persistence manager to store company internal users in the corporate directory, whereas external users are stored in a separate directory.

- **Attribute-based data partitioning**
  
  Different sets of attributes are written to different data sources. For example, global user attributes, such as telephone number, email address, and so on, are written to a corporate directory, whereas SAP-specific data is written to a database.

### 6.2.2 Tivoli Directory Server data source configuration for UME

By following instructions per SAP Note 1015699 you will find a data source example file for use with IBM Tivoli Directory Server. The data source file is a preconfigured XML input file containing UME-specific settings to work with Tivoli Directory Server. These settings can be modified either by uploading a new XML configuration file or by modifying each of the parameters manually via the SAP NetWeaver Application Server Java Administration Tool.

**How to set up the UME configuration XML**

For the configuration of external user repositories (LDAP directory servers), UME stores the appropriate settings in XML configuration files. These files can be modified to meet the specifics of a certain directory server product.

To customize the XML configuration, the first step is to identify the scenario that is to be used (for example, read-only or not) and to extract a default configuration XML for this scenario from the default UME installation.

This default XML contains basic settings for the UME and parameters that are specific to the LDAP directory that is used. For certified vendors, SAP provides a list of the directory-specific values of these latter parameters that have to be adapted in the configuration XML (see SAP Note 1015699 for SAP Java UME Settings with IBM Tivoli Directory Server).

The XML configuration file has several sections. The first section is the Root Level section that contains basic settings of the UME that are independent of the
LDAP directory product. To review and customize Tivoli Directory Server specific values, consider the section within the `<dataSource>` tags that correspond to the LDAP directory.

The only parts of this section that have to be adapted for the specific LDAP directory are `<attributeMapping>` and `<privateSection>`. In the section `<responsibleFor>`, no changes have to be made for the directory, but it provides information needed for configuring the `<attributeMapping>` section.

The `<responsibleFor>` section describes which UME data objects (principal types) will be stored in the LDAP directory. Furthermore, it gives the UME attributes of these data objects that will be stored there.

This section does not have to be changed for the LDAP directory, but it provides useful information for `<privateSection>` and the `<attributeMapping>` section.

Example 6-1 shows the general structure of the UME data source XML configuration file.

**Example 6-1  UME data source XML configuration file**

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- Id: //shared_tc/com.sapall.security/60_SP2_REL/src/_deploy/dist/configuration/shared/dataSourceConfiguration_IBM-Tivoli_deep_not_readonly_db.xml#3 $ from $DateTime: 2003/10/06 14:50:40 $ ($Change: 12628 $) -->
<!DOCTYPE dataSources SYSTEM "dataSourceConfiguration.dtd">
<datasources>
  <datasource id="PRIVATE_DATASOURCE" classname="com.sap.security.core.persistence.datasource.imp.DataBasePersistence"
              isreadonly="false"
              isprimary="true">
    <!-- irrelevant for LDAP directory -->
  </datasource>
  <datasource id="CORP_LDAP" classname="com.sap.security.core.persistence.datasource.imp.LDAPPersistence"
              isreadonly="false"
              isprimary="true">
    <!-- parts that are irrelevant for LDAP directory -->
    <responsiblefor>
      <!-- see below: do not change, but check -->
    </responsiblefor>
    <notresponsiblefor>
      <!-- irrelevant for LDAP directory -->
    </notresponsiblefor>
    <!-- irrelevant for LDAP directory -->
  </datasource>
</datasources>
```
Example 6-2 shows how the `<responsibleFor>` section can look for an LDAP directory that is used as storage for three attributes of accounts, twelve attributes of one name space of users, and one attribute of another name space of users. For the groups, the attributes that are stored in the directory fall into three name spaces.

**Example 6-2  `<responsibleFor>` section**

```xml
<responsibleFor>
    <principal type="account">
        <nameSpace name="com.sap.security.core.usermanagement">
            <attributes>
                <attribute name="j_user"/>
                <attribute name="j_password"/>
                <attribute name="userid"/>
            </attributes>
        </nameSpace>
    </principal>
    <principal type="user">
        <nameSpaces>
            <nameSpace name="com.sap.security.core.usermanagement">
                <attributes>
                    <attribute name="firstname" populateInitially="true"/>
                    <attribute name="displayname" populateInitially="true"/>
                    <attribute name="lastname" populateInitially="true"/>
                    <attribute name="fax"/>
                    <attribute name="email"/>
                    <attribute name="title"/>
                    <attribute name="department"/>
                    <attribute name="description"/>
                    <attribute name="mobile"/>
                    <attribute name="telephone"/>
                </attributes>
            </nameSpace>
        </nameSpaces>
    </principal>
</responsibleFor>
```
Attribute mapping
In the `<responsibleFor>` section, the attributes that are to be stored in the LDAP
directory are defined. In the `<attributeMapping>` section, we define the physical
attributes that the UME attributes should be mapped to. The object classes for
the data objects will be defined in the private section. This section will have to be
adapted to the specific LDAP directory that is used.

In the preconfigured files shipped with the UME, the logical attributes of the UME
are mapped to the physical attributes used for the object class/inetOrgPerson in
the X.500 standard.

The `<attributeMapping>` section has the same structure as the
`<responsibleFor>` section. However, it might contain attributes that were not in
the `<responsibleFor>` section, and there is an additional tag `<physicalAttribute
tagname="...">` within the `<attribute>` tags. The value for the name has to be taken
from the list provided by SAP with SAP Note 1015699. If there is no physical
attribute for a UME attribute (for example, because it is not in the
`<responsibleFor>` section), the value `<null>` should be used.

Example 6-3 shows a sample of an UME data source XML configuration file for
the `<attributeMapping>` section.

Example 6-3  UME data source XML configuration file for the `<attributeMapping>` section

```xml
<attributeMapping>
  <principals>
    <principal type="account">
      <nameSpaces>
        <nameSpace name="com.sap.security.core.usermanagement">
          <attributes>
            <attribute name="j_user">
              <physicalAttribute name="uid"/>
            </attribute>
            <attribute name="j_password">
              <physicalAttribute name="userpassword"/>
            </attribute>
            <attribute name="userid">
              <physicalAttribute name="*null*"/>
            </attribute>
          </attributes>
        </nameSpace>
      </nameSpaces>
    </principal>
    <principal type="user">
      <nameSpaces>
        <nameSpace name="com.sap.security.core.usermanagement">
          ...
        </nameSpace>
      </nameSpaces>
    </principal>
  </principals>
</attributeMapping>
```
<nameSpace name="com.sap.security.core.usermanagement.relation">
  <attributes>
    <attribute name="PRINCIPAL_RELATION_PARENT_ATTRIBUTE">
      <physicalAttribute name="*null*"/>
    </attribute>
  </attributes>
</nameSpace>
</nameSpaces>
</principal>
<principal type="group">
  <nameSpaces>
    <nameSpace name="com.sap.security.core.usermanagement">
      <attributes>
        <attribute name="displayName">
          <physicalAttribute name="displayname"/>
        </attribute>
        <attribute name="description">
          <physicalAttribute name="description"/>
        </attribute>
        <attribute name="uniquename" populateInitially="true">
          <physicalAttribute name="cn"/>
        </attribute>
      </attributes>
    </nameSpace>
    <nameSpace name="com.sap.security.core.usermanagement.relation">
      <attributes>
        <attribute name="PRINCIPAL_RELATION_MEMBER_ATTRIBUTE">
          <physicalAttribute name="member"/>
        </attribute>
        <attribute name="PRINCIPAL_RELATION_PARENT_ATTRIBUTE">
          <physicalAttribute name="*null*"/>
        </attribute>
      </attributes>
    </nameSpace>
    <nameSpace name="com.sap.security.core.bridge">
      <attributes>
        <attribute name="dn">
          <physicalAttribute name="*null*"/>
        </attribute>
      </attributes>
    </nameSpace>
  </nameSpaces>
</principal>
UME XML configuration file private section
The `<privateSection>` contains more attributes describing the LDAP directory. Furthermore, a few attributes do not depend on the LDAP directory, but have to be given anyway. Every time that the UME is updated, check for these attributes, if the values should be changed.

Example 6-4 depicts a sample of the `<privateSection>`.

Example 6-4  Private section sample

```
<privateSection>
  ..Further UME specific parameters
    <ume.ldap.access.server_type>IBM-Tivoli</ume.ldap.access.server_type>
    <ume.ldap.access.user_as_account>true</ume.ldap.access.user_as_account>
  <ume.ldap.access.objectclass.user>inetOrgPerson</ume.ldap.access.objectclass.user>
  <ume.ldap.access.objectclass.uacc>inetOrgPerson</ume.ldap.access.objectclass.uacc>
    <ume.ldap.access.objectclass.grup>groupofnames</ume.ldap.access.objectclass.grup>
    <ume.ldap.access.naming_attribute.user>cn</ume.ldap.access.naming_attribute.user>
    <ume.ldap.access.auxiliary_naming_attribute.user>uid</ume.ldap.access.auxiliary_naming_attribute.user>
    <ume.ldap.access.naming_attribute.uacc>cn</ume.ldap.access.naming_attribute.uacc>
    <ume.ldap.access.auxiliary_naming_attribute.uacc>uid</ume.ldap.access.auxiliary_naming_attribute.uacc>
    <ume.ldap.access.naming_attribute.grup>cn</ume.ldap.access.naming_attribute.grup>
    <ume.ldap.default_group_member.enabled>true</ume.ldap.default_group_member.enabled>
  <ume.ldap.default_group_member>cn=DUMMY_MEMBER_FOR_UME</ume.ldap.default_group_member>
</privateSection>
```

Specific attribute mapping settings for Tivoli Directory Server in XML configuration file

For IBM Tivoli Directory Server, the following parameters have to be set.
**Attribute mapping section**

The user account attributes have to carry the values in Table 6-1.

*Table 6-1  User account attributes*

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>j_user</td>
<td>uid</td>
</tr>
<tr>
<td>j_password</td>
<td>userpassword</td>
</tr>
<tr>
<td>userid</td>
<td><em>null</em></td>
</tr>
<tr>
<td>certificatehash</td>
<td><em>null</em></td>
</tr>
<tr>
<td>javax.servlet.request.X509Certificate</td>
<td>usercertificate</td>
</tr>
<tr>
<td>certificate</td>
<td>usercertificate</td>
</tr>
</tbody>
</table>

The user attributes have to carry the values in Table 6-2.

*Table 6-2  User attributes*

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstname</td>
<td>givenname</td>
</tr>
<tr>
<td>displayname</td>
<td>displayname</td>
</tr>
<tr>
<td>lastname</td>
<td>sn</td>
</tr>
<tr>
<td>fax</td>
<td>facsimiletelephonenumber</td>
</tr>
<tr>
<td>uniquename</td>
<td>uid</td>
</tr>
<tr>
<td>loginid</td>
<td><em>null</em></td>
</tr>
<tr>
<td>email</td>
<td>mail</td>
</tr>
<tr>
<td>mobile</td>
<td>mobile</td>
</tr>
<tr>
<td>telephone</td>
<td>telephonenumber</td>
</tr>
<tr>
<td>department</td>
<td>ou</td>
</tr>
<tr>
<td>description</td>
<td>description</td>
</tr>
<tr>
<td>streetaddress</td>
<td>postaladdress</td>
</tr>
<tr>
<td>pobox</td>
<td>postofficebox</td>
</tr>
<tr>
<td>preferredlanguage</td>
<td>preferredlanguage</td>
</tr>
</tbody>
</table>
The group attributes have to carry the values in Table 6-3.

**Table 6-3  Group attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayName</td>
<td>displayName</td>
</tr>
<tr>
<td>description</td>
<td>description</td>
</tr>
<tr>
<td>uniquename</td>
<td>uniquename</td>
</tr>
<tr>
<td>PRINCIPAL_RELATION_MEMBER_ATTRIBUTE</td>
<td>member</td>
</tr>
<tr>
<td>PRINCIPAL_RELATION_PARENT_ATTRIBUTE</td>
<td><em>null</em></td>
</tr>
<tr>
<td>dn</td>
<td><em>null</em></td>
</tr>
</tbody>
</table>

**Private section**

The UME configuration attributes have to carry the values in Table 6-4.

**Table 6-4  UME configuration attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ume.ldap.access.server_type</td>
<td>IBM-Tivoli</td>
</tr>
<tr>
<td>ume.ldap.access.user_as_account</td>
<td>true</td>
</tr>
<tr>
<td>ume.ldap.access.objectclass.user</td>
<td>inetOrgPerson</td>
</tr>
<tr>
<td>ume.ldap.access.objectclass.uacc</td>
<td>inetOrgPerson</td>
</tr>
<tr>
<td>ume.ldap.access.objectclass.grup</td>
<td>groupofnames</td>
</tr>
<tr>
<td>ume.ldap.access.auxiliary_objectclass.user</td>
<td></td>
</tr>
<tr>
<td>ume.ldap.access.auxiliary_objectclass.uacc</td>
<td></td>
</tr>
<tr>
<td>ume.ldap.access.auxiliary_objectclass.grup</td>
<td></td>
</tr>
<tr>
<td>ume.ldap.access.naming_attribute.user</td>
<td>cn</td>
</tr>
<tr>
<td>ume.ldap.access.auxiliary_naming_attribute.user</td>
<td>uid</td>
</tr>
</tbody>
</table>
### 6.3 Conclusion

This concludes our discussion of the various methods of integration between IBM Security identity management products and SAP solutions. In the next chapter we investigate sample scenarios and best practices.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ume.ldap.access.naming_attribute.uacc</td>
<td>cn</td>
</tr>
<tr>
<td>ume.ldap.access.auxiliary_naming_attribute.uacc</td>
<td>uid</td>
</tr>
<tr>
<td>ume.ldap.access.naming_attribute.grup</td>
<td></td>
</tr>
<tr>
<td>ume.ldap.access.auxiliary_naming_attribute.grup</td>
<td>cn</td>
</tr>
<tr>
<td>ume.ldap.default_group_member.enabled</td>
<td>true</td>
</tr>
<tr>
<td>ume.ldap.default_group_member</td>
<td>cn=DUMMY_MEMBER_FOR_UME</td>
</tr>
</tbody>
</table>

**More information:** See the article “UME Configuration for use with IBM Tivoli Directory Server” in the SAP Developer Network for instructions about how to set up Tivoli Directory Server interoperability with the SAP NetWeaver Application Server Java User Management Engine:


**Even more information:** See the article “SAP and IBM Tivoli Directory Server for z/OS,” published in the SAP Developer Network, which explains how to move your LDAP user management to the highest level of security with Tivoli Directory Server on z/OS (and distributed platforms). The article guides you through preparation, installation, and configuration, and provides you with principals of usage of Tivoli Directory Server on z/OS (and distributed environments).

Identity management use cases

In this chapter we describe common deployment use case scenarios that can be addressed by the IBM Security Identity Management solutions for integration with SAP systems and applications. The use cases are driven by preferred deployment scenarios for SAP environments. We also discuss best practices and recommendations for the Tivoli Identity Manager Adapter for SAP NetWeaver implementation.
7.1 SAP HR driven identity feed

With organizations moving toward globally integrated enterprises with all the benefits of digital identities, businesses can no longer afford ineffective management of the identity life cycle.

As mentioned in previous sections, the management of identities is usually influenced by business cycles, employment cycles, customer relationships, agreements, calendar events, and so on. Conversely, as a geographically distributed organization, the traditional SAP Human Capital Management (HCM) distribution model supports identity synchronization between the head office and regional offices. For example, the head office is responsible for organizational structure and planning, whereas hiring, information maintenance, and termination of employees is the responsibility of the regional offices.

It is becoming imperative that identity management become part of the business process and not be viewed as a stand-alone IT department operation. The SAP HCM (HR) driven identity feed integration is a typical use case scenario that uses all benefits of IBM capabilities to integrate to SAP infrastructure and posture identity management as part of business process, of which the human resources department can take partial ownership.

In that way, not only a full identity life cycle can be achieved, but it is possible to tackle some of the areas that the SAP solution is not addressing, like lack of approval capabilities in CUA environments.
Figure 7-1 shows an example of an SAP HR feed-based SAP ERP user account provisioning implementation. In this example, the HR data is exchanged through ALE/IDOC by using IBM Tivoli Directory Integrator (TDI in Figure 7-1), asynchronously. IBM Tivoli Identity Manager (TIM in Figure 7-1) provides the approval workflow process and the provisioning policy for the user provisioning to the target SAP system based on the information of the HR record to create or update the SAP ERP user account, synchronously.

The scenario as illustrated in Figure 7-1 is meant for a real-time HR feed between an SAP ERP HCM system (SAP HR) and IBM Tivoli Identity Manager (ITIM). From the SAP application side, the SAP ALE protocol is used as a transport vehicle between SAP HR and Tivoli Identity Manager. The integration is achieved using IBM Tivoli Directory Integrator (ITDI) to receive SAP IDOC.
documents, process them, and pass them to Tivoli Identity Manager via the JNDI protocol.

To enable interoperability between the components, configuration steps need to be executed on all three products:

- SAP HCM (HR) system
- Tivoli Directory Integrator Server
- Tivoli Identity Manager Server

In the following section we describe the required configuration to enable the SAP system for ALE/IDOC processing and the AssemblyLine configuration for Tivoli Directory Integrator.

### 7.1.1 SAP IDOC configuration

The HR feed integration relies on sending IDOC from the SAP HR system to a Tivoli Directory Integrator AssemblyLine (AL). The Tivoli Directory Integrator AL uses the ALE/IDOC connector to receive the IDOCs (see 5.4, “IDOC Connector for SAP ERP and SAP NetWeaver AS ABAP” on page 121). For the SAP HR system, the IDOC connector is just one more SAP receiver system expecting IDOC documents.

To send an IDOC between the SAP systems, you must first configure the IDOC path between the SAP systems (in this case, SAP HR and Tivoli Directory Integrator). To do this, transmitter and receiver logical systems need to be configured. Those two logical systems must be created, where SAP HR serves as a transmitter and the Tivoli Directory Integrator serves as a receiver logical system. The transmitter system is then assigned to the client, and the receiving system is assigned to the RFC destination.

Note that in typical cases the transmitter (SAP HR) logical system already exists, and only the logical system for the receiver needs to be created.

For the use of ALE and IDOC on SAP systems, the following configuration steps need to be executed:\(^1\):

1. Create Logical Systems for the transmitter and receiver (transaction BD54).
2. Assign the transmitter to the client (transaction SCC4).
3. Create an RFC destination (transaction SM59).

   The RFC destination is used as an interface between SAP and the Tivoli Directory Integrator. To enable communication between the two logical systems, you must configure the IDOC connector in both SAP HR and Tivoli Directory Integrator.

   \(^1\) For a detailed description see the ALE/IDoc Interface documentation on SAP Help at [http://help.sap.com/saphelp_nw70/helpdata/en/0b/2a6095507d11d18ee90000e8366fc2/frameset.htm](http://help.sap.com/saphelp_nw70/helpdata/en/0b/2a6095507d11d18ee90000e8366fc2/frameset.htm)
Chapter 7. Identity management use cases

4. Create a distribution model (transaction BD54).

The distribution model specifies the transmitter and the receiver, in addition to what data between those two systems will be exchanged. The distribution model requires configuration of the message type used. The model contains information about the sender and the receiver, and the message type to be used. In this scenario, HRMD_A is the message type used. This message type includes all HR data. If it is required to not send all HRMD data, it is possible to distribute only the data that are needed by specifying filter.

5. Generate partner profile (transaction BD54).

The partner profiles are generated automatically by transaction BD64. The model layer will be delivered to both logical systems. After generating an IDOC it can be sent from the transmitter system to the receiver system with HRMD_A data.

6. Activate change pointers (transaction BD50/SE38/BD61).

To enable the master data changes to be distributed automatically, two change pointers need to be activated. The first change pointer needs to be activated in general, and the second change pointer activates the changes for the message type. The change pointers monitor the occurrence of changes, such as hiring a new employee, or a change in the organizational unit. These changes are saved in the RBIDOC report. This report must be activated. For the activation, an SAP Developer Access Key is needed and the object RBIDOC must be registered.

7. Create a background job (transaction SM36).

The RBDMIDOC report will be controlled over a background job. It allows you to choose whether the report will be created immediately, including elapsed time, for another job or for an event or to be executed.

For test purposes, the IDOC can be sent manually using the transaction PFAL within the SAP environment. Here, an object of type person is sent, and then the shipping status is checked. This is a good approach for testing the integration without changing data in the SAP HR system.

7.1.2 Configuring the Tivoli Directory Integrator AssemblyLine

Tivoli Directory Integrator is the link between the SAP HR applications and Tivoli Identity Manager. Tivoli Directory Integrator takes the role for the transformation of the IDOC attributes to the required LDAP attributes used by the Tivoli Identity Manager. The transformation of attributes has the advantage that no change is needed in the data structure of the source SAP system. The attributes from the
source SAP system are copied into a Tivoli Directory Integrator Java container, called a work object, and then sent to the specified target destination in the proper format.

For our scenario you would need to create an Tivoli Directory Integrator AssemblyLine with an IDOC connector in iterate mode and a JNDI connector (pointing to the Tivoli Identity Manager service) in write only mode.

The AssemblyLine is a unidirectional data flow. Note that a bidirectional data flow can only be solved by using a second AssemblyLine. Thus, the Tivoli Directory Integrator can connect to the SAP system and the Tivoli Identity Manager connectors must be configured for each direction. The AssemblyLine is divided into two main parts:

- The feed
- The flow

The feed is responsible for the connection to the SAP system and the processing of IDOCs. Use the standard Tivoli Directory Integrator ALE/IDOC connector to read the IDOC documents and to transfer the data into XML attributes. The connector is used as an XSL parser where the attributes in the Input Map are used for the identity feed.

The flow in this use case example is the connection to Tivoli Identity Manager and converts the attributes of the Work Entry into the attributes of the Conn Entry. The Conn Entry is used as a cache between the Input Map and the Output Map. The Output Map is used for the mapping of the Conn Entry attributes to the LDAP attributes. The LDAP attributes are stored using the JNDI connector in the LDAP directory of the Tivoli Identity Manager.

### 7.1.3 Position-based user provisioning

By using Tivoli Directory Integrator to read SAP HR data and Tivoli Identity Manager to provision the account on SAP systems, that constellation can be used in combination to authorize employees to certain target resources based on their position data in the HR system. When the HR system is configured to provide the data every time that a change occurs, then the data flow can be configured to happen automatically, based on trigger events created in Tivoli Directory Integrator.

To enable role-based provisioning derived from SAP HR data, Tivoli Identity Manager needs to be configured to reflect the company’s organization structure. This includes defining business roles with their associated provisioning policies. Also, Tivoli Identity Manager needs to have at least two services installed that are linked to the Tivoli Directory Integrator ALE/IDOC HR connector and provide
a connection to the SAP application. Performing a reconciliation with the SAP system that hosts the target SAP application loads the available SAP roles into Tivoli Identity Manager. Those roles can then be used to define provisioning policies.

## 7.2 Tivoli Directory Server on z/OS and SAP solutions

As described in 3.3, “IBM Tivoli Directory Server” on page 74, where we introduced the Tivoli Directory Server (TDS) product, and Chapter 6, “IBM Tivoli Directory Server” on page 125, where we discussed interoperability with SAP solutions, Tivoli Directory Server is a reliable solution to store LDAP conform data. It provides capabilities to serve and synchronize user data for SAP applications. IBM Tivoli Directory server is a robust and scalable platform for enterprise security and is available on multiple platforms, including z/OS. Nevertheless, the Tivoli Directory Server implementation on z/OS is different from the Tivoli Directory Server implementation on distributed platforms.

**More information:** Detailed information about IBM Tivoli Directory Server on z/OS is provided in the IBM Redbooks publication *IBM Tivoli Directory Server for z/OS*, SG24-7849.

Guidelines for interoperability between IBM Tivoli Directory Server and z/OS IBM Tivoli Directory Server can be found here:


## 7.2.1 Why choose Tivoli Directory Server for z/OS over a distributed platform

The z/OS Lightweight Directory Access Protocol (LDAP) server, part of IBM Tivoli Directory Server for z/OS, is based on a client/server model that provides client access to an LDAP server. An LDAP directory provides an easy way to maintain directory information in a central location for storage, update, retrieval, and exchange.

Many customers move to the highest security levels not just their core systems but also the services to support those systems, especially for security purposes. Tivoli Directory Server for z/OS supports this intention. That is why Tivoli Directory Server for z/OS supports many z/OS functions, such as Automatic Restart Management (ARM), Workload Management (WLM), and Sysplex Clustering, both file based and DB2 based. z/OS on IBM System z provides the
highest availability, which is fully inherited to Tivoli Directory Server when used on z/OS.

Using Tivoli Directory Server on z/OS provides the following benefits:

- **Performance**
  Tivoli Directory Server for z/OS directory Server is closely located to the application provided on a Linux on System z platform. The impacts are a powerful wireless connection (hipersockets) and consequently a lower risk from the network perspective.

- **Reliability**
  Tivoli Directory Server for z/OS is a reliable service because of the deep integration in System z and z/OS management, both from the maintenance and the operational aspect. There is no special attention needed, because Tivoli Directory Server for z/OS is part of the operating system out of the box.

- **Security**
  All Tivoli Directory Server for z/OS resources (that is, binaries, files, commands, databases, and so on) are protected with the means of IBM RACF® in addition to the LDAP access control lists to protect directory entries. This way a high level of security can be implemented.

- **Simplification**
  Tivoli Directory Server for z/OS can significantly simplify the password change and synchronization with other directories by using Tivoli Directory Integrator. Tivoli Directory Server for z/OS can integrate RACF-defined users. With that technology called *Native Authentication*, the administrator can authenticate the users who do not have to maintain their credentials separately in the directory of Tivoli Directory Server.

When using Tivoli Directory Server for z/OS to integrate with SAP solutions, it would bring these benefits to the SAP landscape.

### 7.2.2 Tivoli Directory Server for z/OS configuration

In this section we discuss the Tivoli Directory Server for z/OS configuration.
System setup and preparation steps
To prepare and set up the system for use with Tivoli Directory Server for z/OS, the following steps need to be considered:

1. Define a TSO user TDSZOS for the Tivoli Directory Server for z/OS installation with RACF authorization Special.
2. Linklist must include SYS1.SIEALNKE.
3. APF authorization must be given to library SYS1.SIEALNKE:
   ```
   RDEF PROGRAM * ADDMEM('SYS1.SIEALNKE' '/') UACC(READ) SETROPTS WHEN(PROGRAM) REFRESH
   ```
4. Define LDAP users LDAPSrv with UID(0) and LDAPGRP:
   ```
   RALTER STARTED LDAPSrv.** STDATA(USER(LDAPSrv)) SETROPTS RACLIST(STARTED) REFRESH
   ```
5. Verify or create High Level Qualifiers for components (used in ds.profile):
   ```
   LE    CEE        SYS1
   LDAP  GLD        SYS1
   DB2   SYS1.DSN.* SYS1.DSN.vxxx (depending on the version)
   SSL   GSK        SYS1
   C     CBC        SYS1
   ```
6. DSCONFIG output dataset TDSZOS.LDAP.JCL will be allocated by DSCONFIG.
7. Verify or create PROCLIB member (we use USER.PROCLIB(LDAPSrv)).
8. Verify or create file systems in z/OS USS:
   ```
   /usr/lpp/ldap    Includes product files.
   /etc/ldap        Includes dsconfig input files.
   /var/ldap        Includes schema and checkpoint files (LDBM) and log files (TDBM).
   ```
Installation of Tivoli Directory Server for z/OS

Take these steps to install Tivoli Directory Server for z/OS:

1. Copy the following DSCONFIG files from the product source /usr/lpp/ldap to /etc/ldap:
   - ds.conf
   - ds.db2.profile
   - ds.envvars
   - ds.profile
   - ds.racf.profile
   - ds.slapd.profile

2. Customize /etc/ldap/ds.profile. The following is from our sample ds.profile:

   ADMINDN = "cn=Admin, o=IBM"
   ADMINPW = <your admin password>
   TDBM_SUFFIX = "o=IBM"
   PROG_SUFFIX = 99change later if required
   LDAPUSRID = LDAPSRV
   LDAPUSRGRP = LDAPGRP
   OUTPUT_DATASET = TDSZOS.LDAPSrv.JCL
   GLDHLQ = SYS1
   GSKHLQ = SYS1
   DSN_SDSNEXITHLQ = SYS1.DSN.V910
   DSN_SDSNLOADHLQ = SYS1.DSN.V910
   DSN_SDNSDBRMHLQ = SYS1.DSN.V910
   DSN_SSID = D942
   DB2_VERSION = V9
   CEEHLQ = SYS1
   CBCHLQ = SYS1

   APF_JOBCARD_1 = //LDAPAPF <enter your jobcard>
   APF_JOBCARD_2 =
   APF_JOBCARD_3 =
   APF_JOBCARD_4 =
   APF_JOBCARD_5 =

   PRGCTRL_JOBCARD_1 = //LDAPPROG <enter your jobcard>
   PRGCTRL_JOBCARD_2 =
   PRGCTRL_JOBCARD_3 =
   PRGCTRL_JOBCARD_4 =
   PRGCTRL_JOBCARD_5 =

   DB2_JOBCARD_1 = //LDAPDB2 <enter your jobcard>
   DB2_JOBCARD_2 =
Chapter 7. Identity management use cases

DB2_JOBCARD_3 = 
DB2_JOBCARD_4 = 
DB2_JOBCARD_5 = 

RACF_JOBCARD_1 = //LDAPRACF <enter your jobcard> 
RACF_JOBCARD_2 = 
RACF_JOBCARD_3 = 
RACF_JOBCARD_4 = 
RACF_JOBCARD_5 = 

SLAPD_PROFILE = /etc/ldap/ds.slapd.profile 
DB2_PROFILE = /etc/ldap/ds.db2.profile 
RACF_PROFILE = /etc/ldap/ds.racf.profile/

Customize your /etc/ldap/ds.racf.profile
LDAPGID = 2
LDAPUID = 1 (change to UID(0) if you receive an error in zFS)

3. Customize your /etc/ldap/ds.db2.profile. The following is from our sample ds.db2.profile:
   DB2_PLAN = DSNACLNI
   DB2_PACKAGE = DSNAOCLI
   DB2_LOCATION = LOC1
   DSN_ATTACH_TYPE = CAF
   TDBM_DB2_USERID = LDAPSRV
   TDBM_DB2_DBNAME = LDAPDB
   TDBM_DB2_STORAGEGROUP = SYSDEFLT

4. Customize /etc/ldap/ds.slapd.profile. We started with the given defaults. Changes can be made later in /etc/ldap/ds.conf.

5. Run the DSCONFIG utility.
   The DSCONFIG command options are:
   dsconfig -i profile_file [-s ds_file] [-a yes|no] [-d debug_level] [-?]
   We used:
   dsconfig -i /etc/ldap/ds.profile -a yes
   You should see the following messages after DSCONFIG command execution:
   GLD2002I Directory Server configuration utility has started.
   GLD2003I Directory Server configuration utility has ended.
Verify that the following members are available in output dataset TDSZOS.LDAPSRV.JCL:

- APF
- DBCLI
- DSCONFIG
- DSENVVAR
- DSNAOINI
- LDAPSRV
- PRGMCTRL
- PROG99
- RACF
- TDBSPUFI

6. RACF Job requirements.

   The first requirement concerns the LDAPSRV profile. When you defined the BPX.SERVER profile in the FACILITY class, the user ID had to have update access to the profile. Any RACF command should only be entered if the BPX.SERVER profile is defined.

   The next requirement, PE to UNIXPRIV, applies to chmod and chown commands on the LDBM, GDBM (file-based), CDBM, and schema backend files and directories. This requirement is not necessary with a DB2-based backend.

   The same applies to all SDBM (RACF as repository) related permits.

7. Run the DB2 Job DSNAOINI.

   DSNAOINI initializes the DB2 subsystem interface to the LDAP started task. In this scenario we use the same DB2 subsystem for both the SAP database and the Directory database, but with different table spaces:

   
   000025 [COMMON]
   000026 MVSDEFAULTSSID=SN63
   000027
   000028 [SN63]
   000029 MVSATTACHTYPE=CAF
   000030 PLANNAME=DSNACLI
   000031
   000032 [COBDBN6]
   000033 AUTOCOMMIT=0
   000034 CONNECTTYPE=1

8. Run the DB2 job DBCLI.

   DBCLI initializes the call-level interface between Tivoli Directory Server for z/OS and DB2:
Starting LDAPSrv

Take these steps to start LDAPSrv:

1. In TSO → SDSF issue `/S LDAPSrv`.

2. Verify the MVS syslog for start execution messages such as the following:

   GLD1012I LDAP server restart registration complete on sys
   GLD1004I LDAP server is ready for requests.
   GLD1059I Listening for requests on ::1 port 389.
   GLD1059I Listening for requests on <IP address of host system> port 389.
   GLD1059I Listening for requests on 10.101.4.4 port 389.
   GLD1059I Listening for requests on 10.101.5.4 port 389.
   GLD1059I Listening for requests on 10.101.6.4 port 389.
   GLD1059I Listening for requests on 10.101.7.4 port 389.
   GLD1059I Listening for requests on 127.0.0.1 port 389.

3. In TSO → SDSF issue `/-SN63 START DB2` to start the DB2 backend.

4. Issue `/-SN63 DIS THREAD(*)` and verify the MVS syslog for messages like the following:

   DSNV401I -SN63 DISPLAY THREAD REPORT FOLLOWS -
   DSNV402I -SN63 ACTIVE THREADS -
   NAME     ST A   REQ ID           AUTHID   PLAN     ASID TOKEN
   DB2CALL  N        1              LDAPSRV           0074     0
   DISPLAY ACTIVE REPORT COMPLETE
   DSN9022I  -SN63 DSNVDT '-DIS THREAD' NORMAL COMPLETION

5. Start any LDAP browser or LDAP command client and connect to your system to review the directory schema.

The LDAP server is shipped with two predefined schema files representing schema. These files are `schema.user.ldif` and `schema.IBM.ldif` and are located in the `/usr/lpp/ldap/etc` directory:

1. Load user schema and IBM schema:

   `ldapmodify -h <IP address of host system> -p 389 -D "cn=admin" -w <your admin password> -f /etc/ldap/schema.user.ldif`
   `ldapmodify -h <IP address of host system> -p 389 -D "cn=admin" -w <your admin password> -f /etc/ldap/schema.IBM.ldif`

   For both loads you should see a message like this:

   modifying entry cn=schema
2. Load the suffix, for example:

```plaintext
dn: o=IBM
objectclass: organization
o: IBM
```

You should see a message like this:

```
adding new entry o=IBM
```

For your environment adapt these settings for the LDAP Directory Information Tree and the required Tivoli Directory Server suffixes to your organization structure and company needs.

3. Use any LDAP browser or command line to verify the modification.

**Setting up LDAP synchronization of users for the SAP AS ABAP user repository**

Read SAP Note 1352688 for instructions about how to obtain the IBM schema extension for SAP AS ABAP. In the attachment to the SAP Note you will find a ZIP archive that contains a file for schema enhancement (see SAP Note 888848) and a file for importing the mapping proposals (see SAP Note 983762):

1. Load the SAP schema:

   ```bash
   ldapmodify -h <IP address of host system> -p 389 -D "cn=admin" -w <your admin password> -f /etc/ldap/SAP_schema_AddOnUM.ldif
   ```

   You should see a message like this:

   ```plaintext
   modifying entry cn=schema
   ```

2. Use an LDAP browser or command line to verify the modifications. You should see object class sapAddOnNum and SAP-specific attribute types (Figure 7-2 and Figure 7-3 on page 157).

![Figure 7-2 LDAP schema extension with SAP schema sapAddOnUM](image)
Other configuration options of Tivoli Directory Server for z/OS are similar to Tivoli Directory Server for distributed platforms. See Chapter 6, “IBM Tivoli Directory Server” on page 125, for more information and options to integrate with SAP solutions.

More information: See the article “SAP and IBM Tivoli Directory Server for z/OS” for details about how to install and configure Tivoli Directory Server on z/OS for use with SAP NetWeaver Application Server ABAP and the SAP NetWeaver Application Server Java User Management Engine:


7.3 Tivoli Directory Server as shared user repository

Tivoli Directory Server can be configured to be used as the back-end data source for the SAP NetWeaver Application Server Java User Management Engine (UME). The advantage of this is that IBM Tivoli Access Manager can also be
configured into the same directory server, thus sharing the same user details across the two applications.

This approach provides the advantage of not having to synchronize two directory servers when attempting to use Tivoli Access Manager to provide authentication and authorization functionality to the SAP back-end system. Figure 7-4 illustrates how this integration can be configured. More information about how to configure Tivoli Access Manager with the SAP NetWeaver Portal can be found in 10.3, “IBM Tivoli Access Manager for e-business integration with SAP NetWeaver AS Java Enterprise Portal Core” on page 241.

![Figure 7-4  Tivoli Directory Server as shared user repository](image)

7.3.1 Configuring SAP NetWeaver AS Java UME for IBM Tivoli Directory Server

The SAP UME can be configured to use an LDAP server as a data source. However, there is no option to set up the UME to use an LDAP directory as data source during installation. Instead, you have to install with a database and configure the UME manually after installation. In this section we provide information about how to set up the UME to use an IBM Tivoli Directory Server as the LDAP directory data source.
Perform the following steps to configure the UME for Tivoli Directory Server:

1. Create a UME configuration XML file containing the appropriate values for Tivoli Directory Server. An overview of the appropriate values for Tivoli Directory Server is provided in 6.2.2, “Tivoli Directory Server data source configuration for UME” on page 131.

   More information: Note that the specific values used in the configuration file might differ depending on your environment (for example, when using user-based data partitioning). Therefore, to quicken the process, there is a sample UME configuration XML file supplied with the IBM Tivoli Access Manager for e-Business integration adapter for SAP NetWeaver Application Server Java at the following website that assumes that Tivoli Directory Server will be used to store user and group information with the remaining information stored in the default database:

   [Website Link]

2. Add the configuration file into the UME configuration tree as follows:

   More information: This step does not apply to SAP NetWeaver Application Server Java 7.1.

   a. Start the Config Tool (if it is not already started). For example, in Windows:

      SAPJ2EEEngine_installation\j2ee\configtool\configtool.bat

   b. Switch to configuration editor mode (click the Configuration Editor icon).

   c. Navigate to **cluster_data** → **server** → **persistent** → **com.sap.security.core.ume.service**.

   d. Switch to edit mode (click the icon to switch between view and edit mode).

   e. Right-click **com.sap.security.core.ume.service** and select **Create sub-node**.

   f. Select **File-entry**.
g. Enter the correct name (depending on whether read-only access is desired):
   - dataSourceConfiguration_tivoli_deep_not_readonly_db.xml
   - dataSourceConfiguration_tivoli_deep_readonly_db.xml.

h. If you are using the sample file, select **Upload** and locate it. Otherwise, enter the configuration text in the area provided.

i. Click **Create**.

j. Switch back to config mode (click the Configuration Editor icon).

3. Activate the UME configuration XML file as per the appropriate version:
   - For SAP NetWeaver Application Server Java 2004 and 7.0 (2004s):
     i. Using the Config Tool, select **UME LDAP data**.
     ii. In the Directory Server tab, select the new configuration file. Click **OK**.
     iii. Enter the appropriate connection details and configuration information.
     iv. If using an SSL connection (which we recommend, see the note above), ensure that the ssl option is selected.
     v. Click **Test connection**.
     vi. Click **Test authentication**.
     vii. Enter the appropriate login credentials and click **authenticate**.
     viii. Apply the changes using File → Apply. Click **OK**.
     ix. Click **OK** again.

   - For SAP NetWeaver Application Server Java 7.1:
     i. Log into the SAP NetWeaver Portal (for example, http://<server>:50000/irj).
     ii. Select the **System Administration** tab.
     iii. Select the **System Configuration** sub tab.
     iv. Under Detailed Navigation, select **UME Configuration**.
     v. Click **Modify Configuration** at the top of the main section.
     vi. Under the Data Source sub tab, upload the configuration file that you created in step 1. Click **Browse**, then select the configuration file on disk, and then click **Upload File**.
     vii. Using the Data Source drop-down list, select the entry that corresponds to the configuration file that you uploaded in the previous step. By default, the entry will be the name of the configuration file, for example:
   
   dataSourceConfiguration_tivoli_deep_not_readonly_db.xml
viii. After selecting the appropriate entry in the Data Source drop-down list, the LDAP Server sub tab will now be visible. Click the **LDAP Server** sub tab.

ix. Under the Connection Data subsection, fill out the LDAP server information as in Table 7-1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Name</td>
<td>The hostname of the LDAP server to be used for the UME configuration.</td>
<td>ldap.example.com</td>
</tr>
<tr>
<td>Server Port</td>
<td>The port by which to access the LDAP server to be used for the UME configuration.</td>
<td>389</td>
</tr>
<tr>
<td>Username</td>
<td>A valid LDAP user that SAP will use to access the Directory Server.</td>
<td>cn=root</td>
</tr>
<tr>
<td>Password</td>
<td>The password corresponding to the provided user.</td>
<td>N/A</td>
</tr>
<tr>
<td>User Path</td>
<td>The LDAP suffix under which users are to be retrieved and stored.</td>
<td>o=ibm,c=us</td>
</tr>
<tr>
<td>Group Path</td>
<td>The LDAP suffix under which groups are to be retrieved and stored.</td>
<td>o=ibm,c=us</td>
</tr>
</tbody>
</table>

x. Having entered the correct LDAP server information, click **Validate Configuration**, and if the configuration is successful, you will see the following message:

Connection test successful

xi. Click **Save All Changes**.

4. Restart the SAP AS-Java cluster.

5. Test the changes by authenticating to the SAP AS-Java user administration application (**http://sapwasjava:port/useradmin**) using credentials stored in Tivoli Directory Server.

### 7.3.2 Configuring Tivoli Directory Server write authority

If WebSEAL is configured to use the same Tivoli Directory Server user and group path as SAP AS-Java, user management should be handled with Tivoli Access Manager tools (for example, **pdadmin**), rather than using SAP AS-Java user administration. Doing so ensures that Tivoli Access Manager users are managed correctly. To ensure that user management is handled only by Tivoli Access Manager, ensure that the read-only configuration file has been used.
The combination of Tivoli Identity Manager Adapter for SAP NetWeaver Application Server Java UME (see 4.3, “Tivoli Identity Manager Adapter for SAP NetWeaver AS Java” on page 102) and Tivoli Identity Manager Adapter for Tivoli Access Manager ensures correct management of a unified user registry.

**Note:** There cannot be any duplication of users and groups between the SAP database and Tivoli Directory Server. This is particularly important for the default users, administrator and guest, and default groups, administrators and guests. Refer to the SAP help website for information about how to configure the location of the default users and groups:

http://help.sap.com/saphelp_nw04/helpdata/en/3f/83df3f3e054e1de1000000a155106/content.htm

**7.4 Tivoli Identity Manager sample scenarios and use cases**

The use case scenarios presented include:

- Stand-alone SAP ABAP target server
- CUA target server
- Multiple SAP ABAP stand-alone target servers
- Stand-alone ABAP target with HR modules
- CUA target environment with HR modules
- Single sign-on password management in a stand-alone setup
- Single sign-on password management with a CUA target
- Account locking extension for CUA and non-CUA setup

**Note:** In our sample scenarios, we refer to the Tivoli Identity Manager Adapter for SAP NetWeaver as the Adapter.
7.4.1 Stand-alone SAP ABAP target server

The Adapter can be deployed in a stand-alone SAP ABAP server setup to administer users directly to this SAP ABAP server. The stand-alone system manages its own user, role, and profile registry. The Adapter communicates directly to the system. Figure 7-5 illustrates how the Adapter is deployed in a stand-alone ABAP environment. The Adapter can be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

![Diagram of Adapter deployment for stand-alone ABAP server]

Figure 7-5  Adapter deployment for stand-alone ABAP server

7.4.2 CUA target server

The Adapter supports the SAP CUA architecture and dynamically determines whether the environment is CUA enabled. The Adapter must be configured against the central CUA master server if enabled. In a CUA architecture you can manage all users from a central master SAP system. This master system is then assigned child systems. Generally, the master system forwards user administration request actions to the child systems, for example, *create user X on child system A*. Also possible is for certain actions to flow in the reverse direction from a child system to the master system. This flow ensures that user account information is synchronized between the central master systems and the child systems within the CUA environment.
Figure 7-6 provides a general architectural overview of the adapter when configured into a CUA environment.

Figure 7-6 Adapter deployment with CUA
7.4.3 Multiple SAP ABAP stand-alone target servers

The Adapter can be deployed in an environment where there might be multiple SAP ABAP servers, each hosting their own user repository. A single instance of the Adapter can be deployed against, and administer users directly to, each ABAP server. Figure 7-7 illustrates this use case, which requires the configuration of multiple Adapter provisioning services in the Tivoli Identity Manager server. The Adapter can be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

Figure 7-7 Adapter deployment for multiple stand-alone SAP ABAP servers
As an alternative, multiple Adapter instances can be used to manage each stand-alone SAP ABAP server. These Adapter instances can be managed by the same Tivoli Identity Manager server. Figure 7-8 illustrates this use case. The Adapter can be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

![Figure 7-8: Multiple adapters for multiple stand-alone ABAP servers](image)

### 7.4.4 Stand-alone ABAP target with HR modules

The Adapter includes an extension feature that enables linking of user accounts to SAP HR Infotype 105 subtypes. Three deployment and configuration components are available to enable the activation of this feature with a standalone ABAP server that includes the HR application modules. These are the components:

- Installation of custom ABAP transport code modules onto the target SAP ABAP server. These transports are included with the Adapter distribution.
- Additional XSL transformation files deployed with the Adapter.
- Configuration of advanced mapping in the Tivoli Identity Manager SAP NetWeaver provisioning service.
The Adapter user guide and release notes describe the necessary deployment and configuration steps in support of the HR linking extension. Figure 7-9 illustrates the Adapter deployment with an ABAP stand-alone server with an HR module environment. The Adapter can be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

Figure 7-9  Adapter with HR link extension for stand-alone ABAP server

### 7.4.5 CUA target environment with HR modules

The HR Linking extension of the Adapter also supports CUA environments where one of the child servers is used as the HR repository. The possibility exists to link user accounts on that child server to the HR records on that child server. The following deployment components are needed to enable this feature:

- Installation of custom ABAP transport code modules onto the child server hosting the HR repository. These transports are included with the Adapter distribution.
- Installation of custom ABAP transport code modules on the CUA master server. These ABAP modules proxy the HR linking requests from the Adapter to the CUA child server.
- RFC destination configuration on the CUA master to the CUA child with HR. This component is needed to enable the ABAP modules on the CUA master to invoke the custom modules on the CUA child server.
- Additional XSL transformation files deployed with the Adapter.
- Configuration of advanced mapping in the Tivoli Identity Manager SAP NetWeaver provisioning service.
- Configuration of RFC destination name in the Tivoli Identity Manager SAP NetWeaver provisioning service.
The Adapter user guide and release notes describe the necessary deployment and configuration steps in support of the HR linking extension. Figure 7-10 illustrates this use case. The Adapter might be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

![Figure 7-10 Adapter with HR link extension for CUA ABAP environment](image)

**7.4.6 Single sign-on password management in a stand-alone setup**

By default, the Adapter uses the standard SAP BAPI RFCs to perform the user management and provisioning functions. However, when these functions are invoked by an external client such as the Adapter, account password changes are flagged as though the administrator has set the password. A consequence of this is that SAP prompts the users to reset their password at their next logon. The single sign-on password management extensions of the Adapter are offered to disable this prompting and set the Tivoli Identity Manager initiated password change as a productive password for user accounts in SAP.

To enable this extension feature, the following deployment components are needed:

- Installation of custom ABAP transport code modules onto the target SAP ABAP server. These transports are included with the Adapter distribution.
- Additional XSL transformation files deployed with the Adapter.
- Configuration of advanced mapping in the Tivoli Identity Manager SAP NetWeaver provisioning service.
Figure 7-11 illustrates the Adapter single sign-on password management in an SAP ABAP stand-alone environment. The Adapter can be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

![Diagram of Adapter with SSO password management](image)

**Figure 7-11 Adapter with SSO password extension for stand-alone ABAP server**

### 7.4.7 Single sign-on password management with a CUA target

Single sign-on password management extensions are supported in a CUA environment. The following deployment components are required:

- Installation of custom ABAP transport code modules onto the CUA master and child ABAP servers. These transports are included with the Adapter distribution.
- Additional XSL transformation files deployed with the Adapter.
- Configuration of advanced mapping in the Tivoli Identity Manager SAP NetWeaver provisioning service.
Figure 7-12 illustrates the use case. The Adapter can be hosted on a server separate from both Tivoli Identity Manager and SAP servers.

Figure 7-12  Adapter with SSO password extension for CUA

7.4.8 Account locking extension for CUA and non-CUA setup

The Adapter supports an extension feature that enables the preservation of account locks, which have been set by an SAP Basis administrator. The extension is supported in CUA and stand-alone SAP deployments. The deployment steps and components are consistent with other extension features described previously. The following deployment components are required:

- Installation of custom ABAP transport code modules onto the CUA master and child ABAP servers. These transports are included with the Adapter distribution.
- Additional XSL transformation files deployed with the Adapter.
- Configuration of advanced mapping in the Tivoli Identity Manager SAP NetWeaver provisioning service.
7.5  Tivoli Identity Manager best practices

In this section, we present known best practices and recommendations related to the Adapter. The discussion includes these topics:

- Deployment of Tivoli Directory Integrator
- Performance issues
- High availability
- A global SAP infrastructure
- SAP Secure Network Communications (SNC)
- SAP version mixture
- Unicode and non-Unicode support
- SAP message server

7.5.1  Deployment of Tivoli Directory Integrator

From the Adapter's perspective, the placement of the Tivoli Directory Integrator is the most important consideration. Note the considerations and best practices in Table 7-2.

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>On which operating system should Tivoli Directory Integrator be deployed?</td>
<td>The adapter supports all platforms that are supported by Tivoli Directory Integrator. The recommendation is deployment on an AIX system to fully take advantage of the optimizing capabilities that it offers.</td>
</tr>
<tr>
<td>An instance of Tivoli Directory Integrator already exists within the enterprise. Can this be reused?</td>
<td>The recommendation is to host Tivoli Directory Integrator on a dedicated server. This practice can help ensure that no other applications are consuming resources over those that the adapter might need.</td>
</tr>
<tr>
<td>We are increasing reliance on VMware hosted machines. Can I deploy Tivoli Directory Integrator into the virtualized environment?</td>
<td>The deployment of Tivoli Directory Integrator can be supported in a virtualized environment, but it is not recommended. Degraded performance has been reported in virtualized environments.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the hardware considerations for hosting Tivoli Directory Integrator based adapters?</td>
<td></td>
</tr>
<tr>
<td><strong>Best practice:</strong> Adapter runtime execution performance benefits with increased access to random access memory and CPU speed. Increase the hard drive size if logging and auditing records are required. Depending on the logging configuration, the adapter can result in many gigabytes of log files.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Consideration:</strong> The SAP landscape is distributed across geographical boundaries. Does it matter where I deploy the Tivoli Directory Integrator instance?</td>
<td></td>
</tr>
<tr>
<td><strong>Best practice:</strong> The recommendation is to host the Tivoli Directory Integrator instance as close to the target SAP ABAP server as possible.</td>
<td></td>
</tr>
</tbody>
</table>
7.5.2 Performance issues

There can be many causes of degraded Tivoli Directory Integrator Adapter performance. Note the considerations and best practices in Table 7-3.

Table 7-3 Considerations and best practices

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have completed a default installation of Tivoli Identity Manager and Tivoli Directory Integrator. What can I do to increase performance?</td>
<td>Default installations do not provide optimal operational performance. Tivoli Identity Manager relies on three major applications stacks: IBM WebSphere Application Server, Tivoli Directory Server, IBM DB2</td>
<td>Each of these can be fine-tuned to achieve optimal performance. Be especially attentive to the performance tuning and indexing of the Tivoli Directory Server because this is where the most performance gains occur.</td>
</tr>
<tr>
<td></td>
<td>Contact each application’s support chain to request information about fine-tuning the respective applications. IBM has specialists to assist with regard to this exact requirement.</td>
<td>Tivoli Directory Integrator can also be configured for performance optimization. Best practices include these: Increase the total number of threads that the Tivoli Directory Integrator can use concurrently, Decrease the trace level enabled. This frees up CPU and input/output resources, ensuring that the adapter operations have a higher priority. Finally, fine-tuning the operating system can also provide performance gains. Of importance is allocating enough memory to the Tivoli Directory Integrator to process all operations without having to move data out to the hard drive. I/O operations are extremely costly on CPU resources, which will impact performance significantly. Depending on the operating system CPU scheduler, a recommendation is to give the Tivoli Directory Integrator process as high a priority as possible.</td>
</tr>
<tr>
<td>I have applied performance tuning to all the IBM applications within my environment but the operations seem to be taking a long time to complete. What is wrong with the adapter?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.5.3 High availability

Ensuring that a system has minimal downtime is critical to many organizations. Establishing and maintaining an infrastructure that supports high availability can be very complex. Note the considerations and best practices in Table 7-4.

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>How can I achieve high availability for my Tivoli Identity Manager deployment?</td>
<td>If you have completed all performance tuning with regard to the IBM products and you continue to observe degraded performance, the recommendation is to performance-tune your SAP NetWeaver server. The adapter execution times significantly rely on response times of SAP RFC calls. IBM is unable to assist with the performance tuning of SAP NetWeaver server. Contact SAP support to assist in this matter.</td>
</tr>
<tr>
<td>How can I achieve high availability for my Tivoli Directory Server deployment?</td>
<td>Tivoli Directory Server offers high-availability functionality through LDAP replication. LDAP replication allows information updated in one server to be propagated to another. Be sure to configure a replication agreement to help ensure that no downtime is caused by the Tivoli Directory Server.</td>
</tr>
<tr>
<td>How can I achieve high availability for my Tivoli Directory Integrator deployment?</td>
<td>The best practice with deploying Tivoli Directory Integrator is to use a load balancer and deploy multiple installations of Tivoli Directory Integrator. The load balancer can distribute the requests to each of the Tivoli Directory Integrator servers. If one of the servers becomes non-responsive, the load balance will no longer send requests to that server.</td>
</tr>
<tr>
<td>How can I achieve high availability for my SAP deployment?</td>
<td>IBM cannot assist with the configuration of SAP in a highly available environment. Contact SAP support for assistance with this matter.</td>
</tr>
</tbody>
</table>
Figure 7-13 illustrates an example of a highly available environment when applying these best practices.

*Figure 7-13 Adapter deployment in a high-availability environment*
7.5.4 A global SAP infrastructure

With more organizations expanding their operations around the world, their IT infrastructure needs to support this. Commonly, a business divides its IT infrastructure into geographical regions. The SAP CUA architecture supports this from a user account management perspective. Figure 7-14 illustrates an example of this SAP infrastructure. In such cases, Tivoli Identity Manager should be deployed to take advantage of the existing CUA infrastructure landscape.
The deployment of Tivoli Identity Manager and the Adapter allows for central administration of users across such an environment. Note the considerations and best practices in Table 7-5.

### Table 7-5 Considerations and best practices

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Consideration:</strong> How can I best deploy Tivoli Identity Manager?</td>
<td><strong>Best practice:</strong> Tivoli Identity Manager can be the central administrative capacity used to administer the user identities across the multiple CUA environments. A strong recommendation in such a complex environment is that you consider both the performance and high-availability details outlined previously.</td>
</tr>
<tr>
<td><strong>Consideration:</strong> How should I create the Tivoli Identity Manager adapter services?</td>
<td><strong>Best practice:</strong> Create an adapter service for each SAP CUA environment. This approach helps ensure that operations completed within one service do not affect others.</td>
</tr>
<tr>
<td><strong>Consideration:</strong> How can I best deploy Tivoli Directory Integrator?</td>
<td><strong>Best practice:</strong> Deploy a Tivoli Directory Integrator server to service each SAP CUA master. This approach helps ensure that operations from one geographical region do not affect the completion or performance of another.</td>
</tr>
</tbody>
</table>
Figure 7-15 illustrates how to deploy the Adapter into a global SAP infrastructure.

Figure 7-15  Example of a global SAP infrastructure deployment with Tivoli Identity Manager
7.5.5 SAP Secure Network Communications (SNC)

SNC is a foundation of the SAP encryption of data communications. Note the consideration and best practice in Table 7-6.

Table 7-6 Considerations and best practices

<table>
<thead>
<tr>
<th>Consideration: Should I configure the adapter to use SNC?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> The adapter is performing as the virtual SAP administrator and presents the same security risk as traditional SAP administration over unsecured communication. For example, new passwords are sent as ordinary parameters over the communication channel. The recommendation is to use SNC with the adapter for SAP communication to encrypt the communication channel.</td>
</tr>
</tbody>
</table>

7.5.6 SAP version mixture

Often an SAP landscape contains a mixture of SAP ABAP server versions (such as 4.5, 4.6, 6.20, 6.40, 7.00, or 7.10). Note the consideration and best practice in Table 7-7.

Table 7-7 Considerations and best practices

<table>
<thead>
<tr>
<th>Consideration: Does the adapter support SAP server version mixtures?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> Yes. However, limitations exist with mixed SAP version support. The adapter release notes describe these limitations. See the adapter documentation.</td>
</tr>
</tbody>
</table>

7.5.7 Unicode and non-Unicode support

Traditionally, SAP instances can be configured to be Unicode or non-Unicode. Note the considerations and best practices in Table 7-8.

Table 7-8 Considerations and best practices

<table>
<thead>
<tr>
<th>Consideration: Should I configure the adapter to connect to a Unicode or non-Unicode SAP server?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> All character data transferred amongst Tivoli Identity Manager Server, the adapter, and the SAP ABAP server are encoded as UTF-8. The recommendation is to configure the adapter against a Unicode SAP server and enable the UTF-8 setting of the adapter.</td>
</tr>
<tr>
<td><strong>Important:</strong> Only provisioning of ASCII characters is supported for non-Unicode SAP ABAP servers. Extended ASCII characters are not tested or supported for a non-Unicode SAP ABAP server.</td>
</tr>
</tbody>
</table>
7.5.8 SAP message server

A common approach in SAP load balancing is to define logon groups and define message servers for those groups. Note the consideration and best practice in Table 7-9.

Table 7-9  Considerations and best practices

<table>
<thead>
<tr>
<th>Consideration: Does the adapter support SAP code pages?</th>
<th>Best practice: No. SAP code pages are not supported.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consideration: What if my SAP landscape has a mixture of Unicode and non-Unicode servers?</td>
<td>Best practice: In a CUA environment the CUA master system must be Unicode. Child systems can be a mixture of both.</td>
</tr>
</tbody>
</table>

7.6 Conclusion

This concludes our discussion of common deployment use case scenarios that can be addressed by the IBM Security Identity Management solutions for integration with SAP systems and applications. This chapter also provided information in regard to best practices to give the reader more insight into how the solutions should be used.
Access management integration

In this part we describe the IBM Security access and entitlement management products and offerings for integration with SAP solutions. We discuss integration points and capabilities that are offered by IBM Security software for access and entitlement management of an SAP environment.
Managing access across IT resources in most cases means to provide technologies that allow you to establish an instance to handle central authentication and single sign-on (SSO) to enterprise systems and applications.

IBM takes a divide-and-conquer approach to this challenge, and addressed different domains with different technologies. These are three domains of authentication and single sign-on:

- **Enterprise or desktop environments**
  
  For SAP environments this is mostly related to SAP GUI used on Windows desktops.

- **Web environments**
  
  A good example for an SAP user interface product to be used in web environments is the SAP NetWeaver Portal.

- **Federated environments**
  
  Federated environments are supported by SAP via SAP Enterprise Services to address service-oriented architectures (SOAs), while authentication and authorization is covered via support of the SAML protocol in most cases.

---

Addressing each of these separately is technically feasible and allows for the realization of true SSO.

SSO for each of the three domains is implemented by the following workings:

- Enterprise SSO, a mechanism to store and manage a user’s account IDs and their associated password or other credentials.
- Web SSO, a mechanism by which HTTP services are aggregated behind the security proxy. The security proxy is responsible for establishing authenticated sessions in a centralized point in the infrastructure.
- Federated SSO, a mechanism by which users can be asserted from one administrative domain to another.

By surveying the different types of SSO and the benefits of each, you will be in a good position to clearly articulate your company’s SSO requirements and to identify a solution that can deliver a full range of SSO capabilities. Figure 8-1 depicts the positioning of the three conceptual types of single sign-on.
Each of the three computing models has single sign-on requirements and IBM applies different technologies to meet each of the SSO requirements. These technologies are implemented in the following products:

- **IBM Tivoli Access Manager for Enterprise Single Sign-on**
  
  This addresses the enterprise SSO problem by deploying an agent on the desktop, which intercepts authentication requests by applications and automatically fills in the login data with credentials stored on the local machine.

- **IBM Tivoli Access Manager for e-business**
  
  This addresses the web SSO problem by placing a reverse web proxy in front of the enterprise web applications. Tivoli Access Manager user accounts are stored in an enterprise directory, and users only need to authenticate to the Tivoli Access Manager server to access all of the existing web applications configured behind the reverse proxy.

- **IBM Tivoli Federated Identity Manager and Tivoli Federated Identity Manager Business Gateway**
  
  This addresses the federated SSO problem by implementing industry standard federated SSO protocols, including SAML, Liberty ID FF, WS-Federation, Information Card, Open ID, and OAuth. It supports arbitrary identity transformations based on XSLT, Tivoli Directory Integrator AssemblyLines, or custom Java programming, so that credentials can be converted to a format compatible with the local environment.
8.1 Tivoli Access Manager for Enterprise Single Sign-On

Tivoli Access Manager for Enterprise Single Sign-On allows organizations to automate access to corporate information, strengthen security, and enforce compliance at the enterprise endpoints.

With Tivoli Access Manager for Enterprise Single Sign-On, organizations can efficiently manage business risk, achieve regulatory compliance, decrease IT costs, and increase user efficiency. Organizations do not have to choose between strong security and convenience.

Tivoli Access Manager for Enterprise Single Sign-On delivers the following capabilities, without requiring changes to the existing IT infrastructure:

- Strong authentication for all user groups
- Enterprise single sign-on with workflow automation
- Comprehensive session management ability
- User-centric access tracking for audit and compliance reporting
- Secure remote access for easy, secure access, any time and anywhere
- Integration with user provisioning technologies


8.1.1 AccessProfiles

Tivoli Access Manager for Enterprise Single Sign-On can provide strong authentication, access automation, and compliance reporting for applications across enterprise endpoints. It can simplify, strengthen, and track access to all Microsoft Windows, web, Java, mainframe, and teletype applications, across all major network access points.

Tivoli Access Manager for Enterprise Single Sign-On AccessProfiles are structured XML files that enable SSO automation for applications. These profiles are created using the Access Studio component, and they are essentially used to automatically capture and inject application credentials located in a user’s Wallet.
8.1.2 Integration with SAP solutions

Tivoli Access Manager for Enterprise Single Sign-On integration with SAP solutions includes the three following aspects:

- Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP GUI
- Single sign-on to SAP GUI for HTML (SAP WebGUI) with Tivoli Access Manager for Enterprise Single Sign-On
- Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP Business Objects XI intranet website

AccessProfile for SAP GUI

IBM provides a Tivoli Access Manager for Enterprise Single Sign-On profile for the SAP GUI. It enables SSO to the SAP GUI by capturing logon credentials on first use, then injecting those stored logon credentials upon subsequent attempts.

The Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP solutions provides the following integration features and capabilities:

- Works with SAP GUI for Windows. Several versions are supported.
- Straightforward and simple setup. No need for complicated configurations with SAP Logon Tickets, SAML, Windows NTLM, or Kerberos.
- Automatic capture of SAP credentials without user intervention.
- Ability to automatic sign-on to SAP systems with previously captured SAP credentials.
- Different SAP systems landscapes supported, such as SAP load balancer, and multiple SAP systems on a single server.
- Supports changing SAP passwords with auto-generated random password.

The Tivoli Access Manager for Enterprise Single Sign-On access profile consists of two distinct parts:

- The first part, called profile_SAP_LOGON, is associated with the saplogon.exe application.
- The second part, called profile_SAP_GUI, is associated with the sapgui.exe application.

See Chapter 9, “IBM Tivoli Access Manager for Enterprise Single Sign-on” on page 197, for more details about the AccessProfile for SAP GUI.
Single sign-on to SAP GUI for HTML

Tivoli Access Manager for Enterprise Single Sign-on Single can be used for sign-on to the SAP GUI for HTML (SAP WebGUI) via the Basic Authentication access control mechanism. Tivoli Access Manager for Enterprise Single Sign-On supports Basic Authentication with two browser profiles, one for Internet Explorer and one for Mozilla Firefox. These two profiles are general web page Basic Authentication credential capture and inject profiles, and are not specific to SAP systems.

See 9.7, “Web single sign-on to SAP GUI for HTML” on page 215, for more details.

AccessProfile for SAP BusinessObjects XI Intranet Website

Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP BusinessObjects XI Intranet Website handles single sign-on automation on the logon and change password screens for SAP Business Objects XI. The site signature for the intranet website needs to be changed with the appropriate URL for the logon and change password web pages.

More details, including download information, for the Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP BusinessObjects XI Intranet Website are available on the IBM Integrated Service Management Library at the following location:


8.2 Tivoli Access Manager for e-business

The Tivoli Access Manager for e-business is a web authentication and authorization solution for corporate web applications. Tivoli Access Manager for e-business allows you to control user access to protected information and resources that are being accessed using a web interface. By providing a centralized, flexible, and scalable access control solution, Tivoli Access Manager for e-business allows you to build secure and easy-to-manage network-based applications and e-business infrastructures.

8.2.1 Components

Tivoli Access Manager for e-business provides a number of core software components to provide authentication and authorization controls. The following
are the components that are important when considering the integration between Tivoli Access Manager for e-business and SAP platforms:

- **Tivoli Access Manager for e-business User Registry**

  The Tivoli Access Manager for e-business includes an LDAP storage that contains all the information associated with the enterprises, including this information:

  - User information
  - Group information
  - Policy information
  - Machine information

  As of Tivoli Access Manager for e-business Version 6.1.1, the following LDAP server implementations are supported:

  - IBM Tivoli Directory Server
  - IBM z/OS LDAP Server
  - IBM Lotus Domino Server
  - Microsoft Active Directory
  - Microsoft Active Directory Application Mode (ADAM)
  - Novell eDirectory
  - Sun Java System Directory Server

- **Tivoli Access Manager for e-business Policy Server**

  The Access Manager Policy Server maintains the master authorization database for the secure domain. This server is primarily used for two types of administrative activities:

  - Modifying the registry to define which objects participate in the secure domain
  - Updating the authorization database with policy definitions

  The Policy Server manages the master authorization database, which, in addition to resource policies, contains location information about other Access Manager servers in the secure domain. Local replicas of the master authorization database are available for resource managers via a push/pull method initiated through the Access Manager runtime service. Each secure domain can only have one Policy Server.

- **Tivoli Access Manager for e-business WebSEAL Server**

  Tivoli Access Manager for e-business has several resource managers that build upon the core infrastructure to provide access control to web-based applications.

  WebSEAL is a high-performance, multi-threaded reverse proxy that resides in front of protected web applications. It applies a security policy to a protected object space. WebSEAL can provide single sign-on solutions and incorporate
back-end web application server resources into its security policy. Because it is implemented on an HTTP server foundation, it is limited to enforcing policy for applications communicating with HTTP and HTTPS protocols.

The combination of the Tivoli Access Manager for e-business user registry, policy server, and WebSEAL provide the following services to ensure that enterprise resources are accessed by legitimate users only:

- **Authentication**
  
  The Tivoli Access Manager for e-business authentication service uses a wide range of built-in authenticators:
  
  - Basic authentication (username and password)
  - Forms-based authentication (username and password)
  - SPNEGO (Kerberos)
  - Client-side certificate
  - HTTP headers
  - IP address
  - Failover cookie
  - CDSSO ID token
  - Token passcode

  In addition, Tivoli Access Manager for e-business provides an *External Authentication Interface (EAI)*. This allows companies to externalize the authentication process of its users to meet their business needs.

- **Authorization services**
  
  The Tivoli Access Manager for e-business authorization service provides coarse-to-fine-grained authorization control using the following policy types:
  
  - Access Control Lists (ACLs)
  - Protected Object Policies (POPs)
  - Authorization Rules (Rules)

  By using these policy types it can be ensured that the organization can allow or deny its users access to IT resources depending on the business requirements.

- **Single sign-on via Junction Resources**
  
  The concept of single sign-on is fairly straightforward. Users log on to a central point but are able to access multiple resources located on different physical servers without having to re-authenticate. Tivoli Access Manager WebSEAL provides this functionality through its junctioning functionality. Tivoli Access Manager WebSEAL junctioning allows connections to be made to backend web application servers. When WebSEAL receives a request, it maps the request to the backend server and provides the user’s ID over to that server. The server has been configured to accept and consume the user ID provided and creates its own representation of the user. After the user has
authenticated to Tivoli Access Manager WebSEAL, he does not need to re-authenticate.

Figure 8-2 illustrates the concept of general single sign-on.

![Diagram showing the concept of general single sign-on](image)

**Figure 8-2  General single sign-on concept**

The general single sign-on concept (Figure 8-2) implies the following generic process flow:

1. The user sends a request to the generic single sign-on server.
2. The generic single sign-on server requests that the user provide some form of identification credentials. After the user provides their credentials, the generic single sign-on server validates them and creates a local representation of the user's identity.
3. The generic single sign-on server passes the request onto the backend server, including a representation of the user's identity.
4. The generic backend server trusts the connection from the generic single sign-on server and extracts the user's identity from the request. The generic backend server then creates a local representation of the user's identity.
5. The generic server then sends the requested resource back to the generic single sign-on server.
6. The generic single sign-on server then forwards the request onto the user's browser, allowing the user to see the requested content.

Tivoli Access Manager for e-business can be integrated into existing and emerging infrastructures to provide secure, centralized policy management capability. Tivoli Access Manager for e-business integrates with IBM WebSphere Application Server, IBM WebSphere Portal, IBM Tivoli Identity Manager, IBM Tivoli Access Manager for Enterprise Single Sign-On, and IBM Tivoli Federated Identity Manager to form a complete Enterprise Identity Management solution.
8.2.2 Integration with SAP solutions

IBM provides the following integrations between Tivoli Access Manager for e-business and SAP solutions:

- IBM Tivoli Access Manager for e-business integration with SAP NetWeaver Application Server ABAP
- IBM Tivoli Access Manager for e-business integration with SAP NetWeaver Application Server Java
- IBM Tivoli Access Manager for e-business integration with SAP NetWeaver Application Server Java Enterprise Portal Core
- IBM Tivoli Access Manager for e-business integration with SAP Internet Transaction Server
- Single Sign-On for SAP NetWeaver Application Server ABAP with Tivoli Access Manager for e-Business in conjunction with SAP NetWeaver Application Server Java

More detailed information about each of the above integrations is located in Chapter 10, “IBM Tivoli Access Manager for e-business” on page 221. That information includes step-by-step instructions for assisting administrators to secure an existing SAP enterprise landscape using Tivoli Access Manager for e-business.

In addition, Chapter 12, “Access management use cases” on page 283, provides extra scenarios, use cases, and a set of best practices with regard to the integration between Tivoli Access Manager for e-business and SAP landscapes.

We recommend that reviewers first read the information within Chapter 10, “IBM Tivoli Access Manager for e-business” on page 221, to give them a good understanding how the products can be integrated. Then read Chapter 12, “Access management use cases” on page 283, to gain more information about how the integrations fit into the enterprise.
8.3 Tivoli Federated Identity Manager

A federation is considered a group of two or more trusted business partners bound by business and technical agreements that allow a user from one federation partner (participating company) to seamlessly access resources from another partner in a secure and trustworthy manner. In a federated business model (in which services are being federated, or shared, between business partners), an organization shares identity data about its users with trusted partners. Sharing identity data enables a partner-organization to obtain information about a third-party identity (for example, customer, supplier, or client employee) from that user's home organization. This approach can eliminate the need for the partner-organization to create and manage identity data for the third-party user.

This federation approach spares a user from having to register at another organization's website and consequently having to remember yet another login ID and password, and can instead use the identity issued by the user's home organization to access the other organization's website and applications. This technique can result in improved integration, communication, and information exchange among suppliers, business partners, and customers, using IT systems and procedures to help lower overall costs, improve productivity, and maximize efficiency in business operations.

Tivoli Federated Identity Manager is a solution that offers federated web single sign-on and allows organizations to participate in a federation. It provides organizations with flexibility by supporting all three major federation standards:

- Liberty
- WS-Federation
- Security Assertion Markup Language (SAML)

Tivoli Federated Identity Manager supports user-centric identities such as OpenID, Information Card Profile using Microsoft CardSpace, and OAuth as identity selectors.

In addition, Tivoli Federated Identity Manager enables compliance reporting in SOA environments.

8.3.1 Federated Identity Management functionality

The Federated Identity Management functionality is built around a trust infrastructure implemented by the Tivoli Federated Identity Manager trust service. This infrastructure is the basis for the Tivoli Federated Identity Manager solutions provided for federated provisioning, federated single sign-on, and Web Services Security management.
Each of these solutions can be deployed independently of the other. Likewise, they can all be deployed in the same environment to provide a complete federation solution.

As shown in Figure 8-3, Tivoli Federated Identity Manager provides overall functionality for these:

- Identity federation
- Federated provisioning
- Web single sign-on
- Web Services Security

![Figure 8-3  Tivoli Federated Identity Manager runtime services](image)

**More information:** To learn more about IBM Tivoli Federated Identity Manager see the following IBM Redbooks publications:

- *Propagating Identity in SOA with Tivoli Federated Identity Manager*, REDP-4354
- *Federated Identity and Trust Management*, REDP-3678
**8.3.2 Integration with SAP solutions**

Tivoli Federated Identity Manager provides access management and single sign-on to federated SAP environments and SAML-enabled SAP applications.

Tivoli Federated Identity Manager integration with SAP NetWeaver Application Server includes these:

- Single sign-on via *SAML Browser Artifact* (SAML integration)
- *Secure Token Service* (STS) for the SAP Login Ticket (SAP Token Trust Module)

See Chapter 11, “IBM Tivoli Federated Identity Manager” on page 259, for detailed descriptions of these integrations.

Additional use case scenarios are described in 12.4, “SSO to SAP NetWeaver AS ABAP with Federated Identity Manager using SAML 2.0” on page 328, and 12.6, “Integrate SAP into SOA by federating the SAP login ticket” on page 343.

**8.4 Conclusion**

This concludes the introduction of the IBM Security access management products and the offerings for integration with SAP systems and applications. In the following chapters we look closer at the different integration concepts that we introduced in this chapter.
IBM Tivoli Access Manager for Enterprise Single Sign-on

For users, having to remember all the passwords that they need for the many systems that they need to access can be a major source of frustration. Often, as a common solution, users write down multiple passwords using a notebook or sticky notes, which can present a serious security threat.

IBM offers Tivoli Access Manager for Enterprise Single Sign-On (TAM E-SSO) to address this source of frustration. Tivoli Access Manager for Enterprise Single Sign-On is capable of automatically injecting users' IDs and passwords into multiple user agents (web browsers, Windows clients, Java clients, and mainframes) without requiring any change to the applications themselves. The net result is that users’ lives are simplified because they need to remember fewer passwords, and their productivity is improved by reducing the downtime caused by lost or forgotten passwords.

The rest of this chapter is written based on the assumption of familiarity with Tivoli Access Manager for Enterprise Single Sign-On.
9.1 Single sign-on considerations

As with any project, it is essential to understand the requirements of that project at the outset. The factors that should be considered when planning to use Tivoli Access Manager for Enterprise Single Sign-On to provide single sign-on to SAP applications include the following:

- How many different SAP systems are being used and what are the names of those systems?
- What languages are being used for accessing the SAP systems?
- How do users access the SAP systems? Are they using the SAP Windows client, or a web browser, or both?
- If using the SAP Windows client, what versions of the client are being used?
- Will SAP application shortcuts be used?
- If using a web browser, what URLs are used for accessing the SAP systems?
- How many fields do users enter when logging on to SAP applications? Is it just username and password, or does the user also need to enter the client name and/or language?
- How should the names of the SAP systems be displayed in the Tivoli Access Manager for Enterprise Single Sign-On Wallet manager?
- Is there any synchronization of the passwords between the different SAP systems? (That is, if the user changes his password on one SAP system does this have any effect on any other SAP system?)

When determining the answers to these questions one should ensure that all likely groups of users are considered. The answers to these questions will affect how exactly Tivoli Access Manager for Enterprise Single Sign-On is best implemented.

9.2 System name display requirements

When the SAP Windows client is launched, a list of systems is displayed. The system names that are initially displayed are the values of the Description fields for the various systems.

**Where are my attributes:** The Description, Server, Database, System, Attribute, and other attributes for the various systems are stored in the `saplogon.ini` file installed with the SAP Windows client on the Windows system.
The user might select, for example, a system whose description is \textit{A1 : ABCD - Production}. The SAP Windows client then establishes a connection with the appropriate system based on information from \texttt{saplogon.ini}.

Alternatively, the user might access the same SAP system through a web browser by specifying a URL of, for example, \url{https://sysabcd.acme.com/}.

A decision needs to be made as to how the system names should be displayed in the Tivoli Access Manager for Enterprise Single Sign-On Wallet manager (the \textit{authentication service} name). The most thought is required if both web browsers and SAP Windows clients are being used by the same users to access the same systems, particularly where the respective names bear little resemblance to each other.

Here are suggested approaches:

- Where the SAP system is accessed solely through the web browser, the authentication service name could correspond to the fully qualified domain name of SAP system URL (such as \texttt{sysabcd.acme.com}).

- Where the SAP system is accessed solely through the SAP Windows client, the authentication service name could correspond to the description field from \texttt{saplogon.ini} (such as \texttt{A1 : ABCD - Production}).

- Where the SAP system is accessed either through the web browser or through the SAP Windows client, the authentication service name should include both the description and the URL (such as \texttt{A1 : ABCD - Production - sysabcd.acme.com}).

How these display name requirements can be met is described below.

\section*{9.3 User education}

Consideration needs to be given to how the SAP users should be educated about the SSO system. The user experience will change, so users need to be informed in advance that this will happen. The approach used will vary from organization to organization. Some will train team leaders who pass the information on to their team members, whereas others will put training videos on an internal website, and yet others will use leaflets or booklets. It is also worth attempting to set user expectations. There might be times when the system works differently from expected. It is also advisable not to launch too many applications at the same time or to change focus during the login process.

\textbf{Beware of the user:} Users sometimes complain if they are not given perfection, forgetting that their use of a system has just become much easier.
### 9.4 Implementation process overview

Assuming that Tivoli Access Manager for Enterprise Single Sign-On has already been implemented for other applications, an overview of the steps involved in implementing Tivoli Access Manager for Enterprise Single Sign-On for SSO to SAP applications can be as follows:

1. Determine the SAP application SSO considerations discussed above.
2. The standard Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP applications can be obtained by downloading the latest Tivoli Access Manager for Enterprise Single Sign-On Enterprise Software Profile Bundle from the IBM support site (component ID 5724N70IF)\(^1\).
4. Make the necessary client workstation configuration changes (as discussed in the Release Notes for the Tivoli Access Manager for Enterprise Single Sign-On SAP profile, which is included in the bundle).
5. Create the authentication services required.
6. Test and tailor the AccessProfiles as required.

### 9.5 Implementing Tivoli Access Manager for Enterprise Single Sign-On

It is worth noting that this guide is not comprehensive. It provides useful pointers, but definitive information is provided in the formal product documentation\(^2\), together with the *Release Notes* and *Cook Book*\(^3\) for the Tivoli Access Manager for Enterprise Single Sign-On profile bundle. In addition, the exact implementation approach depends on the organization’s specific requirements.

In addition, it is essential to have available a specialist who is familiar with Tivoli Access Manager for Enterprise Single Sign-On AccessProfiles, as it might be

---

\(^1\) The direct link to this resource is [https://www.ibm.com/support/docview.wss?uid=swg24029132](https://www.ibm.com/support/docview.wss?uid=swg24029132).


\(^3\) The Release Notes and Cook Book are included in the IBM Tivoli Access Manager for Enterprise Single Sign-On Enterprise Software Profile bundle zip file that can be obtained from the IBM support site (component ID 5724N70IF) currently located here: [https://www.ibm.com/support/docview.wss?uid=swg24029132](https://www.ibm.com/support/docview.wss?uid=swg24029132)
necessary to modify the SAP AccessProfile to meet the organization’s specific requirements.

9.6 How the AccessProfile works

In this section we take a closer look at the Tivoli Access Manager for Enterprise Single Sign-On application profile.

9.6.1 Application process control

The SAP GUI window is the window where SAP logon credentials are entered, and hence the window from which the AccessProfile must capture and inject credentials. Without Tivoli Access Manager for Enterprise Single Sign-On deployed on an SAP client system, the SAP GUI window is displayed as part of the SAP Logon executable process, and not as a separate application or process. There is only ever one instance of the SAP Logon application process, but there can be multiple SAP GUI windows. Tivoli Access Manager for Enterprise Single Sign-On AccessProfiles are initiated when executable processes are started, and not when windows are rendered, so an AccessProfile cannot cater to a process model where an unknown number of windows exist with each window requiring credential capture and injection.

To cater to this process model, the SAP AccessProfile ensures that each SAP GUI window is owned by a separate application process. To do this the SAP Logon application process destroys the SAP GUI window when it appears and then creates an instance of the SAP GUI application process and hence a new instance of the SAP GUI window.

The SAP AccessProfile consists of two distinct parts. The first part, called profile_SAP_LOGON, is associated with the saplogon.exe application, and the second part, called profile_SAP_GUI, is associated with the sapgui.exe application.

SAP client system configuration information is shared between these two parts of the SAP AccessProfile. This configuration is based on the SAP configuration stored in the saplogon.ini file, which is read by the profile_SAP_LOGON part of the SAP AccessProfile.

9.6.2 Number of capture or injection fields

On the SAP GUI logon screen there are typically four entry fields:

- The client field
- The username field
9.6.3 Single sign-on to SAP applications using SAP Logon

Two SAP profiles are supported with Tivoli Access Manager for Enterprise Single Sign-On. Let us take a closer look at those.

**SAP Logon AccessProfile (profile_SAP_LOGON)**

The profile_SAP_LOGON is instantiated when the saplogon.exe application process is started. The profile_SAP_LOGON remains in the state_setup state until the SAP Logon application main window displays (Figure 9-1).

![Figure 9-1 SAP Logon Main Window (source: SAP AG)](image)

When the window appears, the profile_SAP_LOGON transitions to the state_start_sapgui state. During that transition, the profile parses the saplogon.ini file for SAP server connection configuration information and stores that information in a global array.

The profile_SAP_LOGON waits for you to click Log On or the appearance of the main SAP GUI logon window, at which point the profile_SAP_LOGON transitions to the state_close_launched_window state. The profile_SAP_LOGON profile then destroys the main SAP GUI logon window, as that window is not associated with a separate application process. The destruction of that window occurs via the Close A Window action on all triggers, leading to a transition to the state_relaunch state. After a short delay, the profile_SAP_LOGON initiates a script that invokes the SAP GUI application process as a separate instance during the state transition to the state_launched state. In turn, the profile_SAP_GUI is instantiated for that instance of the SAP GUI application process.
**SAP GUI AccessProfile (profile_SAP_GUI)**

The profile_SAP_GUI is instantiated when the `sapgui.exe` (or a similar) application process is started. The profile_SAP_GUI remains in the state_setup state until the SAP GUI main window appears. When that window appears, the profile_SAP_GUI transitions to the SAP_GUI_Logon_Window_Displayed state. During that state transition, a script is run to copy the globally stored SAP configuration information into local storage. Immediately a transition to Was_Stored_Cred_Injected state occurs, during which an injection of the first screen entity is attempted after first fetching credentials from the Wallet. The particular entity injected depends on the number of screen fields supported by the profile (see 9.6.2, “Number of capture or injection fields” on page 201). Depending on whether the injection was successful, the built-in NO_ACCOUNT_DATA_FOUND flag is set or cleared accordingly. At this point, the profile flow follows one of two paths, which join together again at the Creds_Injected state. The following state transition path is followed if credentials were found in the Wallet: **Was_Stored_Cred_Injected → Stored_User_Injected → Stored_Password_Injected → Stored_Language_Injected → Creds_Injected.**

If no credentials were found in the Wallet the state transition path is **Was_Stored_Cred_Injected → Was_Capture_Cancelled → Captured_Client_Injected → Captured_User_Injected → Captured_Password_Injected → Captured_Language_Injected → Creds_Injected.**

The profile_SAP_GUI uses the *Show a dialog to capture logon credentials* action to capture the required entry fields (Figure 9-2).

![Capture Logon Credentials dialog](source:SAP AG)

*Figure 9-2  Capture Logon Credentials dialog (source: SAP AG)*

From the Creds_Injected state, the profile handles the possible error messages that can occur during the logon attempt with the user-entered screen entry field contents.
There are several other windows that can display when in this state. First, the Information window can display. When this occurs, the profile waits for the information window to be canceled before proceeding.

The other window that can display is the password change dialog, which is invoked using the **New Password** button or when the existing password has expired. When this dialog displays, the profile uses the *Show a dialog to capture change password credentials* action to capture the new password (Figure 9-3). The profile caters for errors occurring during the password change.

![Figure 9-3 Change password dialog](image)

Successful logon to the selected SAP system occurs when the main SAP GUI window displays. This window is usually called SAP Easy Access. When this occurs, the profile transitions from the Creds_Injected state to the LoggedOn state. This profile never returns to the start_state state.

### 9.6.4 Single Sign-On to SAP applications using SAP Shortcut

In this section we look at three distinct use cases.

**Launching Shortcut from SAP Logon (profile_SAP_LOGON)**

The profile_SAP_LOGON is instantiated when the saplogon.exe application process is started. The profile_SAP_LOGON remains in the state_setup state until the SAP Logon application main window (Figure 9-1 on page 202) displays. When the window displays, the profile waits for the user to launch a shortcut from the Shortcuts tab. After a shortcut is launched, the profile transitions to the state_after_logon_sapShortcut, during which it executes a script that reads information from the sapshortcut.ini file. It then transitions back to the state state_launched and closes the SAP Shortcut window while launching a new separate process that initiates the shortcut so that the profile_SAP_GUI profile can be instantiated.

**Launching Shortcut from desktop (profile_SAP_LOGON)**

The profile_SAP_LOGON AccessProfile is instantiated when an SAP Shortcut is launched from the desktop. When the profile detects that a shortcut was launched, it transitions to the state state_initial_SAP_Shortcut and closes the
shortcut window while launching it in a separate process, similar to the method described in 9.6.3, “Single sign-on to SAP applications using SAP Logon” on page 202. The profile then transitions to the state_launched state to wait for any user action with the SAP Logon application (that is, connect to other SAP Systems) and handles them as described in the previous sections.

**SAP Shortcut in SAP GUI (profile_SAP_GUI)**

When the profile_SAP_GUI AccessProfile is instantiated and detects that an SAP Shortcut window was launched, it transitions to the state SAP_Shortcut_Window_Displayed. A script is then executed to determine the name of the authentication service to use. Eventually, after injecting and capturing credentials, the profile transitions into the Creds_Injected state. At this point, the profile is handled the same way as described in “SAP GUI AccessProfile (profile_SAP_GUI)”.

### 9.6.5 AccessProfile language support

The AccessProfile for SAP applications supports multiple languages, and each language is specifically handled by the profile. To add additional language support, the profile must be modified. Specifically, the login error messages are language-specific. For example, in the case of an incorrect username and password, the English language error text displayed is Name or password is incorrect (repeat logon), whereas in German it is Name oder Kennwort ist nicht korrekt (Wiederholen Sie die Anmeldung).

There are equivalent error messages for other languages that the SAP system supports. The profile_SAP_GUI uses regular expressions to abbreviate the text required to be entered to these error messages. In the case of the example above, this is the entire regular expression string:

```
(.*incorrect.*)|(.*nicht korrekt.*)|(.*niet correct.*)|(.*incorrectos.*)|(.*errati.*)|(.*correto.*)
```

This handles the error message occurring in English, German, Dutch, Spanish, Italian, and Portuguese.

Appropriate regular expression strings are used for other error messages.
9.6.6 AccessProfile environment and default settings

There are four key SAP client system environment settings that affect the AccessProfile for SAP applications, which are contained within the script in the state_start to state_setup state transition part of the profiles (Figure 9-4 on page 206):

- Location of the saplogon.ini file
- Name and location of the SAP GUI application executable file (SAPGui.exe)
- Location of sapshortcut.ini file
- Name and location of the SAP Shortcut application executable file (sapshcut.exe)

![State Transition from state_start to state_setup state in AccessStudio](image)

**Default location of system environment settings**

The profile assumes that all the settings are located in their default locations. If they are not in the default location, the profile will have to be modified to suit the location. These are the default locations:

- **saplogon.ini** is located in one of the following paths:
  - Environment variable SAP_LOGON_INI
  - %APPDATA%\SAP\Common\saplogon.ini
  - Registry key at HKLM\SOFTWARE\SAP\SAP Shared\SAPSysdir
  - %SYSTEMROOT% directory

  For example:
  C:\Users\AppData\SAP\Common\saplogon.ini

- **SAP GUI application executable file (SAPGui.exe)** is expected here:
  - Registry key at HKLM\SOFTWARE\SAP\SAP Shared\SAPSysdir
  - %PROGRAMFILES%\SAP\FrontEnd\SAPgui\ directory

  For example:
  C:\Program Files\SAP\FrontEnd\SAPgui\SAPGui.exe
sapshortcut.ini is expected to be in the same folder as saplogon.ini:

- Environment variable SAP_LOGON_INI
- %APPDATA%\SAP\Common\saplogon.ini
- Registry key at HKLM\SOFTWARE\SAP\SAP Shared\ SAPSysdir
- %SYSTEMROOT% directory

For example:
C:\Users\AppData\SAP\Common\sapshortcut.ini

SAP Shortcut Launcher application executable file (SAPshcut.exe) is expected here:

- Registry key at HKLM\SOFTWARE\SAP\SAP Shared\ SAPSysdir
- %PROGRAMFILES%\SAP\FrontEnd\SAPgui\ directory

For example:
C:\Program Files\SAP\FrontEnd\SAPgui\SAPshcut.exe
Editing the AccessProfile to customize system environment settings

In certain client systems, these environment settings can be different from the default. In this case, the system environment settings can be customized by editing the AccessProfiles in AccessStudio:

- In AccessStudio, edit the VBScript code in the profile_SAP_LOGON when the state transitions from state_start to state_setup (Figure 9-5).
- Similarly, in AccessStudio, edit the VBScript code in the profile_SAP_GUI when the state transitions from state_start to state_setup.

![Figure 9-5 Editing the script in between state_start to state_setup](image)

9.6.7 Setting up the registry for SAP Logon

To enable the AccessProfile to work properly (as described in the 9.6.1, “Application process control” on page 201), a registry setting for SAP applications has to be modified. There are three ways to perform this task. The first is using AccessAgent's pre-installation task, whereby the registry key is set when installing AccessAgent on the client machine. The second is by manually editing the registry key. The third method is by using the AccessProfile to automatically set it when SAP Logon is launched.
Method 1: Using AccessAgent preinstallation task

To do this:
1. Edit the Reg\DeploymentOptions.reg file in the AccessAgent installation directory.
2. Add the following lines to the file:

   [HKEY_LOCAL_MACHINE\SOFTWARE\SAP\SAPGUI]
   "StartSapLogon"=dword:00000000

Method 2: Manually editing the registry

To do this:
1. Launch Registry Editor:
   - For Windows 7:
     i. Click the Start menu icon in the lower-left corner of your desktop.
     ii. Click the search field above the Start menu button and type regedit into it.
     iii. Click Regedit in the search results to launch the Registry Editor.
   - For Windows XP and Vista:
     i. Click the Start menu button, then click Run.
     ii. Type regedit into the Open text box.
     iii. Click OK in the Run window to launch the Registry Editor.
2. Navigate to the key HKEY_LOCAL_MACHINE\SOFTWARE\SAP\SAPGUI.
3. Create a DWORD value StartSapLogon (if it does not exist).
4. Edit the DWORD value StartSapLogon and set it to 0.
5. Close the Registry Editor.
Method 3: AccessProfile automatic setting on first launch

Follow these steps:

1. Load the AccessProfile profile_SAP_LOGON in AccessStudio.
2. Edit the VBScript that is executed in the state transition from state_setup to state_start_sapgui (Figure 9-6).
3. Change the line in the beginning of the script from isAutomaticRegistrySetting = 0 to isAutomaticRegistrySetting = 1.

Figure 9-6  AccessStudio Editing the script in between state_setup to state_start_sapgui

9.6.8 Capturing credentials when signing in to SAP applications

To capture credentials when first signing in to SAP applications follow these steps:

1. Launch the SAP Logon application and select the appropriate system to sign on to.
2. When the SAP GUI application loads, wait for the Tivoli Access Manager for Enterprise Single Sign-On credentials capture-prompt to appear (Figure 9-2 on page 203).
3. Enter the credentials into the appropriate text fields. Click **OK**.

4. Tivoli Access Manager for Enterprise Single Sign-On prompts you to save the credentials to your Wallet. Click **Yes** to save.

The user should now be logged into the SAP application and the credentials are now saved for subsequent single sign-on to this system.

### 9.6.9 Generating random password during password change

The AccessProfile for SAP GUI comes with the capability to generate a random password, based on your password policy set in the IMS Server. The AccessProfile has an Enterprise authentication service **auth_SAP_random_password_engine** that is uploaded to the IMS Server together with the AccessProfile. Administrators can set the random password generator policy for SAP GUI credentials in the IMS Server under **AccessAdmin → System → Authentication service policies → SAP GUI random password engine**.
Only the password policies are applicable (Figure 9-7). All other authentication policies in the auth_SAP_random_password_engine authentication service do not apply to the SAP GUI AccessProfile.

**How it works**

The AccessProfile runs a JScript when the SAP GUI shows the change password dialog box. This JScript downloads the password policy from the auth_SAP_random_password_engine authentication service in the IMS Server and then generates a new random password based on the policy. It then injects the new random password into the Change password dialog box. With the current design, users have the option to change (that is, delete and re-enter) a new password of their choice. If this behavior is undesirable, it is easy to add an action to click **OK** after the new password is injected. The Click action button
should be added in the profile_SAP_GUI state transition from AA_ChangePwd_Shown to Wait_AAChangePwd_Destroyed.

9.6.10 Authentication service names

The authentication service name is the name that users see in their Wallet. The ideal configuration depends on whether the SAP applications are accessed either through a web browser, the SAP Windows client, or both.

Access via web browser only
Create a validating web browser AccessProfile for single sign-on to the SAP web interface. That is, use the AccessStudio AccessProfile Generator to create an AccessProfile where the screen that appears upon successful logon has been identified. This avoids capturing incorrectly typed credentials.

By default, the authentication service name displayed in the Tivoli Access Manager for Enterprise Single Sign-On Wallet manager is the hostname used to specify the SAP system (such as sysabcd.acme.com from the example above).

If you want to have a different display name then take these steps:
1. Using Tivoli Access Manager for Enterprise Single Sign-On AccessStudio, create an authentication service with the desired display name.
2. Create a server locator for that authentication service that matches the hostname used to specify the SAP system (such as sysabcd.acme.com from the example above).

Access via the Windows SAP client only
By default the authentication service name displayed in the Tivoli Access Manager for Enterprise Single Sign-On Wallet manager is the server name specified in the saplogon.ini file for the specified SAP system.

If you want to have an authentication service name displayed in the Tivoli Access Manager for Enterprise Single Sign-On Wallet manager that matches the description field from saplogon.ini (such as A1 : ABCD - Production from the example above), then edit the AccessProfile according to the instructions in “Changing the authentication service naming in AccessProfile” on page 214.

Access via both web browser and Windows SAP client
By default the authentication service names displayed in the Tivoli Access Manager for Enterprise Single Sign-On Wallet manager are as described above, depending on whether a web browser or the SAP client is used to access the
SAP system. (It is undesirable to have two different entries in the Wallet for the same SAP system.)

If you want to have a display name that includes both the description and the URL (such as A1 : ABCD - Production - sysabcd.acme.com), then take the following steps:

1. Using Tivoli Access Manager for Enterprise Single Sign-On AccessStudio, create an authentication service with the desired display name (such as A1 : ABCD - Production - sysabcd.acme.com).

2. Create a server locator for that authentication service that matches the web browser hostname used to specify the SAP system (such as sysabcd.acme.com from the example above).

3. Create a server locator for that authentication service that matches the authentication service name generated by the AccessProfile when the Windows SAP client is used (such as a server name like sapsys2.acme.com).

4. Alternatively, in place of the previous step, modify the AccessProfile as described in the previous section and create a server locator for that authentication service that matches the description from saplogon.ini (such as A1 : ABCD - Production).

**Changing the authentication service naming in AccessProfile**

The default naming convention used by the AccessProfile is in the following form:

[Description]_[Application Server / Message Server / Router]

You can change the naming convention used by the AccessProfile if required. In profile_SAP_Logon AccessProfile, edit the script that gets executed between state_relaunch state and state_wait_for_launch_again state. At the beginning of the script, locate the following line:

```
strAuthService = global_profile_sap_logon_desc & "_" & strAuthService
```

The statements to the right side of the equals sign = can be replaced with any combination of variable names in Table 9-1. The resulting statement must be a valid VBScript syntax.

You must duplicate the changes to the script that is executed between the state_setup state and SAP_Shortcut_Window_Displayed state found in profile_SAP_GUI AccessProfile.

Table 9-1  Available variables for use with authentication service ID naming

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>global_profile_sap_logon_server</td>
<td>The application server</td>
</tr>
<tr>
<td>global_profile_sap_logon_router</td>
<td>Router to the application and message server</td>
</tr>
<tr>
<td>global_profile_sap_logon_mssysname</td>
<td>3-character abbreviation of SAP system ID</td>
</tr>
<tr>
<td>global_profile_sap_logon_database</td>
<td>SAP database number</td>
</tr>
<tr>
<td>global_profile_sap_logon_mssrvname</td>
<td>Host name of the message server</td>
</tr>
<tr>
<td>global_profile_sap_logon_mssrvport</td>
<td>The port number of the TCP link</td>
</tr>
<tr>
<td>global_profile_sap_logon_desc</td>
<td>Description of an SAP link as it appears in the SAP Logon</td>
</tr>
<tr>
<td>global_profile_sap_logon_origin</td>
<td>Type of entry: MS_SEL_GROUPS, MS_SEL_SERVER or USEREDIT</td>
</tr>
</tbody>
</table>

9.7 Web single sign-on to SAP GUI for HTML

For many SAP system users, the SAP GUI application is the mechanism used to access their SAP systems. However some users, will want to use the SAP GUI for HTML (SAP WebGUI) interface instead.

Single sign-on to the SAP GUI for HTML is also possible with Tivoli Access Manager for Enterprise Single Sign-on with using Basic Authentication access control mechanism. Tivoli Access Manager for Enterprise Single Sign-On supports Basic Authentication with two browser profiles, one for Internet Explorer and one for Mozilla Firefox. These two profiles are general web page Basic Authentication credential capture and inject profiles, and are not specific to SAP systems.
9.7.1 SAP GUI for HTML

The SAP GUI for the HTML URL is in this form:
http://10.150.22.2:8000/sap/bc/gui/sap/its/webgui

Figure 9-8 shows the logon web page.

Clicking **Log On** brings up the Basic Authentication prompt dialog.
9.7.2 Web Single Sign-On using Internet Explorer browser

Figure 9-9 shows the Basic Authentication prompt dialog for Internet Explorer.

![Basic Authentication prompt dialog](image)

Figure 9-9 SAP WebGUI Basic Authentication with Microsoft Internet Explorer

The Tivoli Access Manager for Enterprise Single Sign-On profile for Internet Explorer Basic Authentication is called sso_site_wnd_iexplore.

**Injecting SAP logon credentials using Internet Explorer profile**

From the state_start state, the profile transitions to state_after_inject via a Window is activated trigger, which triggers when the Internet Explorer Basic Authentication prompt dialog displays. During this state transition, stored credentials are injected into the Basic Authentication prompt dialog. There are multiple instances of this trigger in the Internet Explorer profile to handle the Basic Authentication prompt dialog being presented in different languages.

This is the Windows Xpath signature for the English language version of this trigger:

```
/child::wnd[@title~"Connect to (.*)"]/child::wnd[@class_name="Button" and @ctrl_id=1]/parent::wnd[@class_name="#32770"]
```

Click **OK** to submit the injected credentials to the SAP application for authentication. There is no validation of the correct logon with the injected credentials.
Capturing SAP logon credentials using Internet Explorer profile

When you click OK on the Basic Authentication prompt dialog, credentials entered in the Basic Authentication prompt dialog are stored in the Tivoli Access Manager for Enterprise Single Sign-On Wallet. There is no validation of the correct logon with the injected credentials.

The state transition that occurs when you click OK is state_after_inject to state_after_inject.

9.7.3 Authentication service name for Internet Explorer profile

The authentication service used by the Internet Explorer profile is derived from the webpage URL that being connected to. The Connect To dialog contains the IP address/hostname of the page being connected to in the dialog window title, and the profile uses that part of the URL only for the authentication service name. For the example shown above, the authentication service name would be 10.150.22.2.

9.7.4 Web Single Sign-On using Firefox browser

Figure 9-10 shows the Basic Authentication prompt dialog for Firefox.

![Figure 9-10 SAP WebGUI Basic Authentication with Mozilla Firefox](image)

The Tivoli Access Manager for Enterprise Single Sign-On profile Firefox Basic Authentication is called sso_site_wnd_firefox.

The User Name and Password fields of this dialog are not able to be specified using a Windows Xpath, so the Firefox profile uses a Show a dialog to capture logon credentials action to prompt for and capture logon credentials.

Injecting SAP logon credentials using Firefox profile

From the state_start state, the profile transitions to Basic Auth Shown via a Window is activated trigger, which triggers when the Firefox Basic Authentication
prompt dialog displays. During this state transition, stored credentials are injected into the Basic Authentication prompt dialog. There is only one instance of this trigger in the Firefox profile to handle the Basic Authentication prompt dialog being presented in English.

This is the Windows Xpath signature for this trigger:

/child::wnd[@title="Authentication Required" and @class_name="MozillaDialogClass"]

Click OK to submit the injected credentials to the SAP application for authentication. There is no validation of the correct logon with the injected credentials.

Capturing SAP logon credentials using Firefox profile

From the state_start state, the profile transitions to Basic Auth Shown via a Window is activated trigger, which triggers when the Firefox Basic Authentication prompt dialog displays. Credential injection is attempted, but if credentials are found in the Wallet, then the transition from Basic Auth Shown state to the Basic Auth Inject User state actions a Show a dialog to capture logon credentials action to prompt for and capture logon credentials.

When OK on the “Tivoli Access Manager for Enterprise Single Sign-On AccessAgent” prompt dialog is clicked, the entered credentials are copied to the Basic Authentication prompt dialog, which are stored in the Tivoli Access Manager for Enterprise Single Sign-On Wallet. There is no validation of correct logon with the injected credentials.

Authentication service name for Firefox profile

The authentication service used by the Firefox profile is derived from the webpage URL being connected to. The Authentication Required dialog contains the IP address/hostname of the page being connected to in the dialog window text, and the profile uses that part of the URL and port component of the URL for the authentication service name. For the example shown above, the authentication service name would be 10.150.22.2:8000.

9.8 Conclusion

This concludes our discussion of the IBM Tivoli Access Manager for Enterprise Single Sign-On product and its integrations with SAP solutions. You can find additional use cases in 12.8, “Tivoli Access Manager for Enterprise Single Sign-on SAP use cases” on page 377.
In the next chapter we look at IBM Tivoli Access Manager for e-business and its integration options with SAP solutions.
IBM Tivoli Access Manager for e-business

This chapter provides an overview of the integrations between IBM Tivoli Access Manager for e-business and SAP NetWeaver Application Server-based systems.

The following items are covered:

- “Integration with SAP NetWeaver AS ABAP” on page 222
- “Integration with SAP NetWeaver AS Java” on page 228
- “IBM Tivoli Access Manager for e-business integration with SAP NetWeaver AS Java Enterprise Portal Core” on page 241
- “Tivoli Access Manager for e-business Integration with SAP Internet Transaction Server” on page 243
- “Single sign-on for SAP NetWeaver AS ABAP with WebSEAL in conjunction with SAP NetWeaver AS Java” on page 243

Each of the integrations is broken up into individual sections within the chapter. Each section provides detailed information and instructions about how to successfully deploy the integration.
10.1 Integration with SAP NetWeaver AS ABAP

IBM provides an integration that allows IBM Tivoli Access Manager for e-business to achieve both single sign-on and single sign-off capability with SAP NetWeaver Application Server ABAP.

The integration uses the Tivoli Access Manager WebSEAL component as a reverse-proxy. A WebSEAL junction is created for the SAP NetWeaver Application Server ABAP. Client requests for SAP NetWeaver Application Server ABAP applications are sent through WebSEAL, which prompts the user to provide single sign-on credentials for authentication. When authenticated, WebSEAL retrieves the client's SAP NetWeaver Application Server ABAP credentials from their global sign-on resource and forwards them in a basic authentication header along with the initial request to SAP NetWeaver Application Server ABAP. The SAP NetWeaver Application Server ABAP authenticates the credentials against its user registry and, if successful, returns the requested content back to WebSEAL, which in turn forwards the response onto the user's browser.

Figure 10-1 on page 223 illustrates an overview of the architecture for Tivoli Access Manager WebSEAL integration with SAP NetWeaver AS ABAP.

Figure 10-1 on page 223 shows the integration architecture where the following processes occur:

1. A browser request to SAP NetWeaver Application Server ABAP is submitted through WebSEAL.
2. WebSEAL intercepts the request, authenticates, and authorizes the user as required.
3. WebSEAL retrieves the global sign-on credentials of the authenticated user from the resource allocated to the SAP NetWeaver Application Server ABAP junction.
4. WebSEAL forwards the request to SAP NetWeaver Application Server ABAP with the global sign-on credentials in a basic authentication header.
5. SAP NetWeaver Application Server ABAP authenticates the user against his user registry.
6. If authentication is successful, SAP NetWeaver Application Server ABAP returns the requested content.
7. The content is returned to the browser. WebSEAL performs filtering as appropriate for the junction method.
To achieve single sign-on between Tivoli Access Manager WebSEAL and SAP NetWeaver Application Server ABAP, we need to configure Tivoli Access Manager to store and supply the user credential for SAP NetWeaver Application Server ABAP applications upon user access. The Tivoli Access Manager global sign-on function can be used to perform such a job.

**GSO implications:** Global sing-on mapping can introduce administration overheads and management issues.

As an alternative, you can configure single sign-on for SAP NetWeaver Application Server ABAP applications by leveraging the single sign-on solution for IBM Tivoli Access Manager WebSEAL and SAP NetWeaver Application Server Java in conjunction with the SAP logon ticket.

For more information see 10.5, “Single sign-on for SAP NetWeaver AS ABAP with WebSEAL in conjunction with SAP NetWeaver AS Java” on page 243.
10.1.1 WebSEAL junctions to SAP NetWeaver AS ABAP

To connect WebSEAL with SAP NetWeaver Application Server ABAP, a Tivoli Access Manager WebSEAL junction must be created.

The Tivoli Access Manager WebSEAL junction can be configured to use either TCP or SSL (recommended), depending on the protocol configured for SAP NetWeaver Application Server ABAP. The selected protocol and port number must be the same for both Tivoli Access Manager WebSEAL and SAP NetWeaver Application Server ABAP.

There are two types of junction that can be created to achieve this integration:

- Standard junction
- Virtual host junction

The details for creating such junctions are detailed in the sections that follow.

**Transparent path junction:** Transparent path junctions are not supported in this integration. This is due to SAP inserting Base64 encoded session management information at the root of the path in the URL. This prevents a transparent path junction solution because the root path is never the same value.

For complete details about Tivoli Access Manager WebSEAL junction creation, see the *IBM Tivoli Access Manager WebSEAL Administration Guide*¹.

**Standard junction**

The use of a standard junction causes the inclusion of the junction name in the URL when accessing ABAP applications using the browser. For example:

http[s]://webSEAL.company.com/stdsapjct/sap/bc/gui/sap/its/webgui/

Where `stdsapjct` is the name of the junction.

This junction allows Tivoli Access Manager WebSEAL to identify the back-end server to which requests should be forwarded. As a result, every requested page that the SAP NetWeaver Application Server ABAP sends back to the client via WebSEAL must be filtered. This allows the junction name to be inserted into all the static and dynamically generated links in both the HTML and JavaScript.

¹ The IBM Tivoli Access Manager WebSEAL Administration Guide can be found at the following website:

Below is the template for the `pdadmin` command to create a standard junction (enter the command as one line):

```
pdadmin> server task default-webseald-server_name create
    -t tcp | ssl
    -h sapasabap_fqdn
    -p sap_http[s]_port_no
    -b gso
    -T gso_resource_name
    -j
    /junction_name
```

**Virtual host junction**

Tivoli Access Manager WebSEAL supports virtual hosting and, through virtual host junctions, can eliminate the limitations of URL filtering. Virtual host junctions allow Tivoli Access Manager WebSEAL to communicate with local or remote virtual hosts. Tivoli Access Manager WebSEAL uses the HTTP host header in client requests to direct those requests to the appropriate document spaces located on junctioned servers or on the local machine.

**Note:** Virtual hosting introduces some Domain Name Services and session management challenges. For details, refer to the *Tivoli Access Manager WebSEAL Administration Guide*.

Below is the template for the `pdadmin` command to create a virtual host junction (enter the command as one line):

```
pdadmin> server task default-webseald-server_name virtualhost create
    -t tcp | ssl
    -h sapasabap_fqdn
    -p sap_http[s]_port_no
    -b gso
    -T gso_resource_name
    jct_name
```

To configure Tivoli Access Manager WebSEAL to listen on the same ports as SAP NetWeaver Application Server ABAP, an entry must be added to the [interfaces] stanza of the WebSEAL configuration.

As an alternative, SAP NetWeaver Application Server ABAP can be configured to listen on the same ports as Tivoli Access Manager WebSEAL, thereby eliminating the need to add an interface entry to the Tivoli Access Manager WebSEAL configuration file.
10.1.2 Configuring Tivoli Access Manager WebSEAL options

Due to the nature of the SAP NetWeaver Application Server ABAP integration, a number of different ABAP applications can be accessed using the same junction.

In the event that multiple junctioned resources use cookies with the same name, Tivoli Access Manager WebSEAL inserts the junction name into the cookie name to differentiate them from each other. In addition, the cookie path is set to / thus ensuring that the cookie gets sent to Tivoli Access Manager WebSEAL with each request. The consequence of setting the cookie path to / is that cookies from one ABAP application could be sent to another ABAP application from the same junction, possibly causing the ABAP application to behave unexpectedly or to fail. To prevent this we need to set the following option in the Tivoli Access Manager WebSEAL configuration file:

```[preserve-cookie-names]
mangle-path-into-cookie-name=yes```

10.1.3 Configuring SAP NetWeaver AS ABAP

When a user signs out of an SAP NetWeaver Application Server ABAP application, he must additionally end his session on Tivoli Access Manager WebSEAL. This is achieved by configuring the SAP NetWeaver Application Server ABAP application to redirect to Tivoli Access Manager WebSEAL's pkmslogout page.

The following steps describe the process required to achieve this:

1. From the SAP GUI run the **SICF** transaction.
2. Click **Execute**.
3. Double-click the node in the tree that represents the service that you want to configure.
4. Click **Change**.
5. Select the **Error Pages** tab, then the subsequent **Logoff Page** tab.
6. Select the **Redirect to URL** radio button, then enter the URL of the pkmslogout application on Tivoli Access Manager WebSEAL. For example:

   - **Standard junction**
     
     http://webseal_fqdn/pkmslogout

     Where *webseal_fqdn* is the FQDN that resolves to the WebSEAL host between the client and the SAP server.

   - **Virtual host junction**
     
     http://sapnwasabp_fqdn:port_no/pkmslogout

     Where *sapnwasabp_fqdn* is the FQDN for the SAP server that is routed to Tivoli Access Manager WebSEAL first, and *port_no* is the port where the SAP server listens for requests. (Tivoli Access Manager WebSEAL inherently listens on the same port.)

7. Click **Save** to apply changes to the server.

### 10.1.4 Testing the integration

The WebGUI ABAP application must be activated to test this integration. See the SAP documentation for details about how to configure the SAP WebGUI (SAP GUI for HTML). To test the integration follow these steps.

1. Open a browser and access the SAP NetWeaver Application Server ABAP with the appropriate URL:

   - **Standard junction**
     
     http[s]://webseal_hostname/jct_name/sap/bc/gui/sap/its/webgui/!

     For example:

     http://webseal.company.com/stdjct/sap/bc/gui/sap/its/webgui/!

   - **Virtual host junction**
     
     http[s]://sapasabap_hostname:port_no/sap/bc/gui/sap/its/webgui/!

     For example:

     http://sapserver.company.com:8000/sap/bc/gui/sap/its/webgui/!

2. An authentication request is received from the Tivoli Access Manager WebSEAL. Log in using the Tivoli Access Manager user ID and password.

3. Upon successful authentication, the SAP NetWeaver Application Server ABAP application page should be displayed.
10.2 Integration with SAP NetWeaver AS Java

SAP NetWeaver Application Server Java provides an open infrastructure for deploying J2EE web applications. IBM offers an integration between IBM Tivoli Access Manager for e-business and SAP NetWeaver Application Server Java. This integration provides single sign-on functionality for J2EE web applications deployed on SAP NetWeaver Application Server Java.

The Tivoli Access Manager WebSEAL component is used as a reverse-proxy in front of SAP NetWeaver Application Server Java. Tivoli Access Manager WebSEAL acts as a security gateway that authenticates and authorizes user access. J2EE web applications deployed on SAP NetWeaver Application Server Java are configured to use the User Management Engine.
Users access SAP NetWeaver Application Server Java resources through Tivoli Access Manager WebSEAL. If the user has not yet accessed a resource, she is prompted to provide a user ID, which is authenticated and authorized against the Tivoli Access Manager user registry. If authentication has been successful, the user ID is passed over a Tivoli Access Manager WebSEAL junction to SAP NetWeaver Application Server Java as an HTTP header. SAP NetWeaver Application Server Java is configured to accept and trust this user ID. The user is seamlessly signed on to the J2EE web application (Figure 10-2).

**Figure 10-2  SAP NetWeaver Application Server Java integration architecture**

Figure 10-2 shows the integration architecture, where the following processes occur:

1. A client uses his browser to access SAP NetWeaver Application Server Java through Tivoli Access Manager WebSEAL.
2. Tivoli Access Manager WebSEAL intercepts the request, authenticates, and authorizes the user.
3. On successful authentication, Tivoli Access Manager WebSEAL passes the request to SAP NetWeaver Application Server Java, together with the username in the form of an HTTP header.
4. SAP NetWeaver Application Server Java is configured to read the HTTP header, and the user is authenticated to the J2EE web application.
5. Optionally, the login module stack creates an SAP login ticket to be used by other SAP applications. The login ticket is passed back to the browser.
10.2.1 WebSEAL junctions to SAP NetWeaver AS Java

A Tivoli Access Manager WebSEAL junction must be created to connect Tivoli Access Manager WebSEAL with SAP NetWeaver Application Server Java. The Tivoli Access Manager WebSEAL junction can be configured to use either TCP or SSL (recommended).

For this integration, the junction creation command must specify the `-c iv_user` option, which configures Tivoli Access Manager WebSEAL to send the authenticated username in the iv-user HTTP header.

There are three types of junction that can be created to achieve this integration:

- Virtual host junction
- Transparent path junction
- Standard junction

The details to create such junctions are described in the sections that follow.

For complete details about Tivoli Access Manager WebSEAL junction creation, see the *IBM Tivoli Access Manager WebSEAL Administration Guide*.

Virtual host junction

The use of virtual host junctions eliminates the limitations of URL filtering. Virtual host junctions allow Tivoli Access Manager WebSEAL to communicate with local or remote virtual hosts. Tivoli Access Manager WebSEAL uses the HTTP host header in client requests to direct those requests to the appropriate document spaces located on junctioned servers or on the local machine.

**Note:** Virtual hosting introduces some domain name services and session management challenges. For details, see the *Tivoli Access Manager WebSEAL Administration Guide*. 
Below is the template for the `pdadmin` command to create the virtual host junction (example command, entered as one line):

```
pdadmin> server task instance-webseald-server_name virtualhost create -t tcp | ssl -h sapas-java_fqdn -p port_no -c iv_user junction
```

See the *Tivoli Access Manager WebSEAL Administration Guide* for more details on virtual host junctions.

**Watch your ports:** By default, SAP NetWeaver Application Server Java is configured to listen on port 50000+instance_number. For example, instance 99 listens on port 50099.

Therefore, to ensure that the virtual host junction works correctly, take one of the following steps:

- Configure Tivoli Access Manager WebSEAL and SAP NetWeaver Application Server Java to use the same ports.
- Configure Tivoli Access Manager WebSEAL with an interface listening on the same ports as SAP NetWeaver Application Server Java.

**Transparent path junction**

Links to resources located on the backend server must be filtered by Tivoli Access Manager WebSEAL to ensure that the user does not access it directly, but rather through the Tivoli Access Manager WebSEAL junction. There are three sections of a URL that need to be considered:

- Protocol (http/https)
- Hostname:port
- Path

Transparent path junctions remove the need to filter the path.

A transparent path junction observes a crucial requirement—the configured junction name must match the name of a subdirectory under the root of the back-end server document space. All resources accessed through this junction must be located under this subdirectory. The transparent path junction name represents the name of the actual subdirectory on the back-end server.

For SAP NetWeaver Application Server Java, the number of root directories that need to be accessed depend on the J2EE web application. Create a transparent path junction to each root directory used by the J2EE web application.
To create a transparent path junction, use the same command as for creating a standard junction but include the \(-x\) option. The main difference between a standard and transparent junction is the fact that the junction name is not stripped when Tivoli Access Manager WebSEAL passes the request to the back-end SAP NetWeaver Application Server Java.

Using the SAP User Administration web application as an example, enter the commands below (entered as one line):

```
pdadmin> server task instance-webseald-server_name create
    -t tcp
    -c iv_user
    -h sapas-java_fqdn
    -p port_no
    -x /useradmin

pdadmin> server task instance-webseald-server_name create
    -t tcp
    -c iv_user
    -h sapas-java_fqdn
    -p port_no
    -x /logon
```

For more details on transparent path junctions, refer to the *Tivoli Access Manager WebSEAL Administration Guide*.

**Standard junction**

Below is the template for the `pdadmin` command to create a standard junction (entered as one line):

```
pdadmin> server task instance-webseald-server_name create
    -t tcp
    -h sapas-java_fqdn
    -p port_no
    -c iv_user /junction_name
```

For more detailed instructions on WebSEAL junction creation, refer to the *IBM Tivoli Access Manager WebSEAL Administration Guide*.

**Protocol switching**

With the current implementation of this integration there is a known Microsoft Internet Explorer browsers limitation known as protocol switching.
When accessing backend applications through Tivoli Access Manager WebSEAL, there are two distinct connections:

- A connection from the browser to Tivoli Access Manager WebSEAL
- A connection from Tivoli Access Manager WebSEAL to the SAP NetWeaver Application Server Java application

Each connection can use a different protocol, for example, HTTP or HTTPS.

Protocol switching occurs when the protocol used by the connection from the browser to Tivoli Access Manager WebSEAL is different from the protocol used by the connection to the SAP NetWeaver Application Server Java application. For example, protocol switching from SSL to TCP occurs when the browser is accessing Tivoli Access Manager WebSEAL using SSL (HTTPS) and the Tivoli Access Manager WebSEAL junction to SAP NetWeaver Application Server Java is created using TCP (-t tcp).

If your environment and user base are affected by protocol switching, download the latest version of the integration from the following URL and follow the instructions outlined under the section titled “Protocol switching (optional)”: http://www.ibm.com/support/docview.wss?uid=swg24007296

### 10.2.2 Junction Mapping Table (JMT)

Server-relative URLs generated on the client-side by applets and scripts initially lack knowledge of the junction point. Tivoli Access Manager WebSEAL cannot filter such URLs because they are generated dynamically on the client side. During a client request for a resource using such a URL, Tivoli Access Manager WebSEAL can attempt to reprocess the server-relative URL using a pre-defined mapping table.

This document does not provide any specific steps for configuring Tivoli Access Manager WebSEAL to correctly filter content from back-end SAP NetWeaver Application Server applications. The steps below provide an overview of the steps required to create and configure the JMT, should it be required.

Perform the following tasks to create and configure a junction mapping table:

1. Open the Tivoli Access Manager WebSEAL configuration file:
   ```
   webseal_install_dir/etc/webseald-instance.conf
   ```
2. Within the `[junction]` stanza, set the value of the `jmt-map` option to this:
   ```
   lib/jmt.conf
   ```
3. If there is no previously created jmt.conf file, create the file in this directory:
webseal_install_dir/www-instance/lib

4. Modify the webseal_install_dir/www-instance/lib/jmt.conf file, creating a mapping table for the appropriate URL pattern. For example, using the SAP User Administration web application and assuming that your junction name to the J2EE Web application is /jct_sapas-java, the following lines should be created:
   /jct_sapas_java /useradmin/*
   /jct_sapas_java /logon/*

5. Reload the Junction Mapping Table. From pdadmin, enter this command:
   pdadmin> server task webseald-server_namejmt load

Other application junction mapping table settings must be worked out by the integration implementer to meet the specific application requirements.

Note: For additional information about configuring the Junction Mapping Table, refer to the Tivoli Access Manager WebSEAL Administration Guide.

10.2.3 Configuring Tivoli Access Manager WebSEAL options

A Tivoli Access Manager WebSEAL option must be added to allow SAP NetWeaver Application Server Java applications to utilize SAP single sign-on, using an SAP login ticket.

Note: This option is not required if the SAP applications are not configured for SAP single sign-on.

Modify the webseal_install_dir/etc/webseald-instance.conf file by adding the following entry to the [preserve-cookie-names] stanza:
name = MYSAPSSO2

10.2.4 Configuring the Tivoli Access Manager WebSEAL logout page

When a user successfully logs on to Tivoli Access Manager WebSEAL, and in turn through single sign-on to the SAP NetWeaver Application Server Java, session cookies from the SAP NetWeaver Application Server Java are stored in the browser's memory. Unless the user physically closes the browser, the session between the browser and SAP NetWeaver Application Server Java remains open, even if the user has logged out of Tivoli Access Manager. If another user were to re-use the same browser window to log on to Tivoli Access
Manager WebSEAL and access the SAP NetWeaver Application Server Java, the SAP NetWeaver Application Server Java might assume that the new user was the same as the previous user.

To overcome this issue complete the following steps:

1. Make a backup copy of your existing logout page:
   `webSEAL_install_dir/www/lib/html/locale/logout.html`

2. Use an editor to open the `logout.html` file in the integration package.

3. Copy the script element from the HTML code (contained between the script start and end tags):
   ```html
   <SCRIPT language="Javascript">
   ......
   </SCRIPT>
   ```

4. Paste this code into your existing `logout.html` file.

5. In the body tag, add this parameter
   ```html
   <BODY bgColor="#FFFFFF" text="#000000"
   onLoad="delete_all_cookies('/', exception_list)"
   >
   ```
   For example:
   ```html
   <BODY bgColor="#FFFFFF" text="#000000"
   onLoad="delete_all_cookies('/', exception_list)"
   >
   ```


   This additional JavaScript code ensures that all cookies are deleted, including the MYSAPSSO2 cookie, when the users logs out of Tivoli Access Manager WebSEAL. This ensures that if another user logs in using the same browser, that she is unable to impersonate the initially logged-in user. You can provide an `exception_list` that can contain cookie names that you do not want deleted. All other cookies that are not within this list will be deleted by the above JavaScript.

### 10.2.5 SAP NetWeaver AS Java configuration

To make this integration work, additional configuration is required for SAP NetWeaver Application Server Java. This includes the steps described in the following sections.

**Note:** The following steps target SAP NetWeaver Application Server Java 7.1. There might be differences between the newer and older versions.
Adjusting the login module stack to use header variables

When a user is authenticated on SAP NetWeaver Application Server Java, the server processes the stack of login modules that apply to the J2EE web application that the user accesses. The header variable login module is not automatically included with the default login module stacks. Therefore, to use header variables for authentication, you must adjust the login module stacks for those applications that will use header variables to authenticate a user:

1. Create the HeaderVariableLoginModule as per the appropriate version:
   a. In the SAP NetWeaver Administrator, click **Configuration Management**.
   b. In the Security sub-tab, select **Authentication**.
   c. Select the **Login Modules** tab.
   d. Click **Edit** in the top right, which allows changes to be made.
   e. Under Login Modules, click **Add**.
   f. Fill in the required fields as shown in Table 10-1.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name</td>
<td>com.sap.security.core.server.jaas.HeadersVariableLoginModule</td>
</tr>
<tr>
<td>Display Name</td>
<td>HeaderVariableLoginModule</td>
</tr>
</tbody>
</table>

   g. Click **OK**, and then **Save** in the top right.

2. Adjust the J2EE web application's login module stack by adding the HeaderVariableLoginModule as per the appropriate version:
   a. In the SAP NetWeaver Administrator, click **Configuration Management**.
   b. In the Security sub-tab, select **Authentication**.
   c. On the Components tab, click **Edit** in the top right, which allows changes to be made.
   d. Under Component Policy Configurations, select the **ticket** item.
   e. Under Details for Selected Component, on the Authentication Stack tab, click **Add**.
   f. From the Select Login Module drop-down list, select **HeaderVariableLoginModule**, and then click **OK**.
   g. Select **HeaderVariableLoginModule** from the list that is displayed next, then change its flag value to **REQUIRED**.
h. Under the Options for the Selected Login Module section, add options for the HeaderVariableLoginModule by clicking **Add**, and enter the values as listed in Table 10-2.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ume.configuration.active</td>
<td>TRUE</td>
</tr>
<tr>
<td>Header</td>
<td>iv-user</td>
</tr>
</tbody>
</table>

i. Move the HeaderVariableLoginModule entry in the list to the top by selecting **HeaderVariableLoginModule** and clicking **Move Up**.

### Adjusting JSessionId cookie from domain to host only cookie

Where there is more than one Java server using JSESSIONID and also using Tivoli Access Manager WebSEAL as a reverse proxy, the generation of the JSESSIONID cookie must be adjusted so that it is not a domain cookie. This is achieved by amending the settings in the `web-j2ee-engine.xml` file as described below.

**Session cookie information:** This problem would manifest itself in the Java server of the application being unable to interpret the JSESSIONID that is being passed or used. For additional information regarding the JSESSIONID cookies when multiple J2EE servers are involved, refer to SAP Notes 791765 and 1144722, which can be found through the SAP Service Marketplace:

http://service.sap.com

Access to the SAP Service Marketplace requires a valid user ID.

Adjust the cookie as follows:

1. Start the SAP NetWeaver Config Tool. For example, in Windows:
   
   SAPJ2EEEngine_installation\j2ee\configtool\configtool.bat

2. Switch to configuration editor mode by clicking the **Configuration Editor** icon.

3. Navigate to the `web-j2ee-engine.xml` file by selecting **cluster_config → system → custom_global → cfg → services, servlet_jsp → persistent → web-j2ee-engine.xml**.

4. Switch to edit mode by clicking the icon to switch between view and edit mode, and double-click the `web-j2ee-engine.xml` file.

5. Above the last tag `</web-j2ee-engine>` insert this:

   ```xml
   <cookie-config>
   <cookie>
   ```
6. Switch back to config tool mode by clicking the Configuration Editor icon again.

**Altering the password change functionality**

By default, the SAP NetWeaver Application Server Java UME security policy forces users to change their SAP password when the password has been created by an SAP administrator, including passwords created for new SAP users. As a result, when authenticating via single sign-on, the user is forced to authenticate to SAP NetWeaver Application Server Java, after successful authentication to Tivoli Access Manager WebSEAL, to change the password set by the SAP administrator.

This might not be a desirable behavior, particularly in situations where all access to SAP NetWeaver Application Server Java applications is via single sign-on. Therefore, configure the SAP UME to not require password changes:

1. Start the Config Tool, located here:
   SAPJ2EEEngine_installation\j2ee\configtool\configtool.bat

2. Click **Switch to configuration edit mode** at the top.

3. Navigate to **cluster_config → system → custom_global → cfg → services → com.sap.security.core.ume.service → Propertiesheet properties**.

4. Switch to edit mode by clicking the switch between the View and Edit mode icons in the top left, and double-click the **Propertiesheet properties** entry.

5. Locate the **ume.logon.force_password_change_on_sso** key and double-click it.

6. In the custom-value field, enter false, and click **Apply custom → OK**.

7. Switch to view mode by clicking the switch between the View and Edit mode icons in the top left.
Customizing the SAP NetWeaver AS Java logout

The default SAP NetWeaver Application Server Java UME logout function must be customized to achieve single sign-off from both SAP Application Server Java and Tivoli Access Manager WebSEAL. Single sign-off only works for the SAP NetWeaver Portal. To achieve this, perform the following steps:

1. Start the Config Tool, located here:
   
   `SAPJ2EEEngine_installation\j2ee\configtool\configtool.bat`

2. Click **Switch to configuration edit mode** at the top of the page.

3. Navigate to **cluster_config → system → custom_global → cfg → services → com.sap.security.core.ume.service → Propertysheet properties**.

4. Switch to edit mode by clicking the switch between the View and Edit mode icons in the top left, and double-click the **Propertysheet properties** entry.

5. Locate the **ume.logoff.redirect.url** key and double-click it.

6. In the custom-value field, enter these values:
   - For transparent path and virtual host junctions, enter the value `/pkmslogout`.
   - For standard junctions, enter this value:
     
     `../pkmslogout`

7. Click **Apply custom**, and then **OK**.

8. Switch to view mode by clicking the switch between View and Edit mode in the top left.

10.2.6 Restarting the SAP NetWeaver AS Java cluster

Restart the SAP NetWeaver Application Server Java cluster for the changes made with the configuration tool to take effect.
10.2.7 Testing the integration

To test the integration, take these steps:

1. Ensure that there is no direct access to the SAP NetWeaver Application Server Java machine. This can be done by adding an entry in the local hosts file, redirecting any references to the SAP NetWeaver Application Server Java machine to Tivoli Access Manager WebSEAL.

2. Open a browser and access the SAP NetWeaver Application Server Java through Tivoli Access Manager WebSEAL:
   - Standard junction:
     \[ http[s]://webseal_fqdn/junction/application \]
     For example (using the useradmin example):
     \[ http://webseal.example.com/jct_sapas_java/useradmin \]
   - Transparent path junction:
     \[ http[s]://webseal_fqdn/application \]
     For example (using the SAP useradmin example):
     \[ http://webseal.example.com/irj \]
   - Virtual host junction:
     \[ http[s]://sapas-java_fqdn:port/application \]
     For example (using the SAP useradmin example):
     \[ http://sapas-java.example.com:50000/useradmin \]

3. An authentication request is received from the Tivoli Access Manager WebSEAL. Log in using the Tivoli Access Manager user ID and password.

4. Upon successful authentication, the J2EE Web application main page displays.

See the integration package and the integration guide for more detailed information: More detailed information about IBM Tivoli Access Manager for e-Business and the integration with SAP NetWeaver Application Server Java can be found at the IBM Support Portal.

Find the IBM Tivoli Access Manager for e-Business integration adapter for SAP NetWeaver Application Server Java here:

10.3 IBM Tivoli Access Manager for e-business integration with SAP NetWeaver AS Java Enterprise Portal Core

This section provides additional steps to integrate Tivoli Access Manager for e-business with SAP NetWeaver Application Server Java Enterprise Portal Core.

**Note:** Before completing this section ensure that all steps have been followed as outlined in 10.2, “Integration with SAP NetWeaver AS Java” on page 228. It is important to ensure that the integration with the base SAP NetWeaver Application Server Java is working before completing the steps in this section.

### 10.3.1 Creating a Tivoli Access Manager WebSEAL Junction

When creating a standard or transparent path junction these must be named `/irj`.

Here is an example command (entered as one line):

```
pdadmin> server task webseald-server_name create -t tcp -h sapep_hostname -p port_no -c iv_user
```

There are no further requirements when creating a virtual host junction for SAP NetWeaver Enterprise Portal Core.

### 10.3.2 Tivoli Access Manager WebSEAL JMT setup

When using a standard junction, a Junction Mapping Table entry is required. Specify the following URL pattern:

```
/irj/*
```

For example, assuming that your junction name to the SAP NetWeaver Application Server Java Enterprise Portal Core is `/sapportal`, the following line should be created:

```
/sapportal /irj/*
```
10.3.3 Tivoli Access Manager WebSEAL configuration options

To allow Tivoli Access Manager WebSEAL to correctly filter the content from SAP NetWeaver Java Enterprise Portal Core, several Tivoli Access Manager WebSEAL options have to be added and updated.

Modify the `webseal_install_dir/etc/webseald-instance.conf` file as follows:

1. Within the `[filter-content-types]` stanza, add the following option to the existing list:
   
   ```
   type = text/xml
   ```

2. Within the `[filter-request-headers]` stanza, add the following option:
   
   ```
   header = accept-encoding
   ```

3. Within the `[script-filtering]` stanza, set the `script-filter` option to the value `yes`. For example:
   
   ```
   script-filter = yes
   ```

4. Within the `[session]` stanza, set the `ssl-id-sessions` option to the value `no`. For example:
   
   ```
   ssl-id-sessions = no
   ```

5. Within the `[filter-url]` stanza, add the following values in the appropriate alphabetic location:
   
   ```
   TREENODE = IMAGEURL
   TREENODE = FOLDERCLOSEIMAGEURL
   TREENODE = FOLDEROPENIMAGEURL
   TREEUPDATE = FOLDERCLOSEIMAGEURL
   TREEUPDATE = FOLDEROPENIMAGEURL
   ```

6. Within the `[server]` stanza, set the `process-root-requests` option to the value `never`. For example:
   
   ```
   process-root-requests = never
   ```

7. Save the file.

8. Restart Tivoli Access Manager WebSEAL.

10.3.4 Configuring SAP NetWeaver AS Java Enterprise Portal Core

For SAP NetWeaver Application Server Java Enterprise Portal Core, be sure to modify the ticket authentication template to include the `HeaderVariableLoginModule`. For guidance about this setting, refer to 10.2.5, “SAP NetWeaver AS Java configuration” on page 235.
10.4 Tivoli Access Manager for e-business Integration with SAP Internet Transaction Server

IBM has deprecated its officially supported integration guide between IBM Tivoli Access Manager WebSEAL and SAP Internet Transaction Server. This decision was made because as of SAP NetWeaver 2004 and onwards, the stand-alone SAP Internet Transaction Server has been integrated into the SAP NetWeaver Application Server as an Internet Communication Framework service.

This means that to configure single sign-on between Tivoli Access Manager WebSEAL and the integrated SAP Internet Communication Framework service, the officially supported method is to follow the *IBM Tivoli Access Manager for e-business SAP NetWeaver ABAP Integration Guide*, which was outlined in 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222.

10.5 Single sign-on for SAP NetWeaver AS ABAP with WebSEAL in conjunction with SAP NetWeaver AS Java

This section describes how to configure single sign-on for SAP NetWeaver Application Server ABAP applications by leveraging the single sign-on solution for IBM Tivoli Access Manager WebSEAL and SAP NetWeaver Application Server Java in conjunction with the SAP logon ticket. This is achieved without the requirement of visible redirections and the Tivoli Access Manager global sign-on lockbox.

10.5.1 Introduction

The Tivoli Access Manager WebSEAL single sign-on solution for SAP NetWeaver Application Server Java is simple to configure and easily managed, as documented in 10.2, “Integration with SAP NetWeaver AS Java” on page 228. The Tivoli Access Manager WebSEAL single sign-on solution for SAP NetWeaver Application Server ABAP, however, requires the use of the Tivoli Access Manager global sign-on lockbox, as it does now allow for a method of single sign-on that is based on trust, except when using the SAP logon ticket. Basic integration information for SAP NetWeaver Application Server ABAP can be found in 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222. The requirement for the use of the global sign-on lockbox becomes cumbersome when using a combination of both SAP NetWeaver Application Server ABAP and Java, as it adds another user registry requiring password synchronization and management.
This section describes how to configure a method using Tivoli Access Manager WebSEAL single sign-on for Sap NetWeaver Application Server ABAP that does not rely on the global sign-on lockbox, but instead makes use of the SAP logon ticket generated from a Tivoli Access Manager WebSEAL single sign-on solution for SAP NetWeaver Application Server Java.

10.5.2 Scenario

Consider an environment where SAP applications are deployed using both the SAP NetWeaver Application Server ABAP and Java. In the current SAP landscapes this is a likely scenario, for example, where a customer has deployed an SAP CRM solution, which is written using ABAP technology, and SAP NetWeaver Application Server Java Enterprise Portal, which is written using Java technology.
Given the limitation of single sign-on options in the SAP NetWeaver Application Server ABAP environment that are based on trust, the identity and access management administrator is required to configure the Tivoli Access Manager global sign-on lockbox to provide login credentials to the SAP NetWeaver Application Server ABAP (Figure 10-3).

This is the process flow for this scenario:

1. The user requests an SAP NetWeaver Application Server ABAP resource via Tivoli Access Manager WebSEAL. Authentication credentials are provided to Tivoli Access Manager WebSEAL, if required.

2. After successful Tivoli Access Manager WebSEAL authentication, Tivoli Access Manager WebSEAL retrieves the user’s SAP NetWeaver Application Server ABAP credentials from the Tivoli Access Manager global sign-on lockbox.

3. The SAP NetWeaver Application Server ABAP credentials are sent to SAP NetWeaver Application Server ABAP using a basic authentication header when passed across the junction from Tivoli Access Manager WebSEAL.
4. SAP NetWeaver Application Server ABAP authenticates the user against the SAP user registry using the credentials extracted from the basic authentication header provided.

5. After successful authentication, the user is supplied with the requested SAP NetWeaver Application Server ABAP content, along with the SAP logon ticket.

6. At a later time, the SAP logon ticket can be used to authenticate to the SAP NetWeaver Application Server Java.

**Note:** It is required to synchronize the usernames and passwords between the global sign-on lockbox and the SAP User Registry.

To avoid the requirement of using the Tivoli Access Manager global sign-on lockbox, and the synchronization of the lockbox credentials with those within the SAP user registry, it is possible for the admin to configure Tivoli Access Manager WebSEAL for redirection on authentication to force the user to access an SAP NetWeaver Application Server Java application first to obtain the SAP logon ticket. The SAP NetWeaver Application Server ABAP application would be configured for SAP logon ticket authentication, and the user would be provided with a link to the SAP NetWeaver Application Server ABAP application from the SAP NetWeaver Application Server Java. Authentication in the SAP NetWeaver Application Server ABAP application would take place using the SAP logon ticket obtained for initial redirect to the SAP NetWeaver Application Server Java.
Figure 10-4 provides an illustration of this configuration.

This is the process flow for this scenario:

1. The user requests an SAP NetWeaver Application Server ABAP resource via Tivoli Access Manager WebSEAL. Authentication credentials are provided to WebSEAL.

2. After successful Tivoli Access Manager WebSEAL authentication, Tivoli Access Manager WebSEAL redirects the browser to an SAP NetWeaver Application Server Java application to obtain the SAP logon ticket.

3. The browser automatically follows the redirect sent by the Tivoli Access Manager WebSEAL.

4. Tivoli Access Manager WebSEAL receives the request for an SAP NetWeaver Application Server Java resource and sends the username of the authenticated user in the form of an HTTP header called iv-user.
5. The SAP NetWeaver Application Server Java receives the request and verifies the existence of the user in the SAP user registry.

6. After successful verification, the user is supplied with the SAP logon ticket, and content from the SAP NetWeaver Application Server Java application is returned.

7. The fact that the content of the SAP NetWeaver Application Server Java is returned might confuse the user. In most cases the user requests the originally requested resource from the SAP NetWeaver Application Server ABAP. This time, the request is not redirected by Tivoli Access Manager WebSEAL because the user is already authenticated.

8. The request for the SAP NetWeaver Application Server ABAP resource is passed to the SAP NetWeaver Application Server ABAP server. In this case the request contains the SAP logon ticket as a cookie generated by the SAP NetWeaver Application Server Java.

9. The SAP logon ticket is used to authenticate the user.

10. After successful authentication, the user is supplied with the requested content.

Unfortunately, although we have alleviated the admin of the burden of managing the global sign-on lockbox, the scenario can become confusing to some users when they are constantly redirected after authentication to an application other than the one that was requested.
10.5.3 Solution

The ideal solution allows for the user to access the desired SAP application, whether it is located on the SAP NetWeaver Application Server ABAP or the Java system, without the confusion of redirections, in addition to removing the requirement to manage the global sign-on lockbox. The section outlines the solution that makes this possible. The solution makes use of the HTTP specification to generate the SAP logon ticket from the SAP NetWeaver Application Server Java in order to be used by the SAP NetWeaver Application Server ABAP in manner that is invisible to the user. Figure 10-5 illustrates an overview of the solution.

![Diagram of solution](image-url)

*Figure 10-5 Single sign-on to SAP NetWeaver AS ABAP and SAP NetWeaver AS Java without visible redirection and without using Tivoli Access Manager global sign-on*
This is the process flow for this scenario:

1. The user requests a resource located on the SAP NetWeaver Application Server ABAP via Tivoli Access Manager WebSEAL. Authentication credentials are provided to Tivoli Access Manager WebSEAL.

2. After successful Tivoli Access Manager WebSEAL authentication, Tivoli Access Manager WebSEAL redirects the browser using a page that contains a link to a hidden image on the SAP NetWeaver Application Server Java and a redirect to the originally requested resource. By containing a link to SAP NetWeaver Application Server Java, the browser obtains the SAP logon ticket.

3. The browser automatically requests the hidden image.

4. Tivoli Access Manager WebSEAL receives the request for the SAP NetWeaver Application Server Java image and sends the username of the authenticated user in the form of an HTTP header called `iv-user`.

5. The SAP NetWeaver Application Server Java receives the request and verifies the existence of the user in the SAP user registry.

6. After successful verification, the user is supplied with the SAP logon ticket and the requested hidden image.

7. At this point, the browser automatically follows the redirect back to the SAP NetWeaver Application Server application. This time, the request is not redirected by Tivoli Access Manager WebSEAL because the user is already authenticated.

8. The request for the SAP NetWeaver Application Server ABAP resource is passed to the SAP NetWeaver Application Server ABAP server. The request contains the SAP logon ticket generated by the SAP NetWeaver Application Server Java.

9. The SAP logon ticket is extracted from the request and used to authenticate the user within the SAP NetWeaver Application Server for ABAP server.

10. After successful authentication, the user is supplied with the originally requested content.

### 10.5.4 Configuring Tivoli Access Manager WebSEAL

This section provides the steps required to configure Tivoli Access Manager WebSEAL.
Configuring redirection on authentication

Perform the following actions to configure Tivoli Access Manager WebSEAL to instruct the browser to retrieve the SAP logon ticket before redirecting to the requested resource after successful authentication:

1. Open the Tivoli Access Manager WebSEAL configuration file.
   For example, with a default installation of Tivoli Access Manager WebSEAL, the configuration file is located at `WebSEAL_root/etc/webseald-default.conf`.

2. Within the `[acnt-mgt]` stanza, locate the `login-redirect-page` parameter.

3. Set the `login-redirect-page` parameter to `/redirect.html`.

4. Within the `[enable-redirects]` stanza, enable the appropriate authentication mechanism. For example, when using forms authentication, ensure that the redirect parameter is set to `forms-auth`.

5. Within the `[server]` stanza, locate the `process-root-requests` parameter.

6. Set the `process-root-requests` parameter to either `filter` or `always`.

7. If the `process-root-requests` parameter is set to `always`, ensure that an entry is added in the `process-root-filter` stanza, for example, `root = /redirect.html`.

8. Restart Tivoli Access Manager WebSEAL to make the above changes take effect.

In summary, the configuration file should contain the settings show in Example 10-1.

*Example 10-1  Tivoli Access Manager WebSEAL Configuration settings*

```plaintext
[acnt-mgt]
...
login-redirect-page = /redirect.html

[enable-redirects]
redirect = forms-auth

[server]
...
process-root-requests = filter
```

Note: Only forms authentication has been validated using the solution described in this publication.
Modifying the Tivoli Access Manager WebSEAL login page
Perform the following actions to modify the Tivoli Access Manager WebSEAL login page to capture the requested URL:

1. Locate the Tivoli Access Manager WebSEAL login page template in the installation’s lib directory.
   For example, with a default installation of Tivoli Access Manager WebSEAL, the lib directory is located here:
   WebSEAL_root/www-default/lib/html/language
   Where language is C for English.
2. Modify the file to contain the JavaScript supplied in Example 10-2.

   **Example 10-2   Redirection page**
   ```html
   <script>
   document.cookie = 'RedirectURL=%HTTP_BASE%%URL%';
   </script>
   ```

Creating the redirection page
Perform the following actions to create the redirection page returned by Tivoli Access Manager WebSEAL after authentication.

1. Create a new file called redirect.html in the Tivoli Access Manager WebSEAL docs directory.
   For example, with a default installation, the docs directory is located here:
   WebSEAL_root/www-default/docs
2. Modify the file to contain the HTML supplied in Example 10-3.

   **Example 10-3   Redirection page**
   ```html
   <html>
   <head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <meta http-equiv="Pragma" content="no-cache">
   <meta http-equiv="Expires" content="-1">
   
   <title></title>
   </head>
   ```
Creating the Tivoli Access Manager WebSEAL junctions

Tivoli Access Manager WebSEAL requires a junction to be created for each of the Application Servers (that is, a junction for the SAP NetWeaver Application Server Java and a junction for the SAP NetWeaver Application Server ABAP).

For assistance with regard to creating the junction from Tivoli Access Manager WebSEAL to the SAP NetWeaver Application Server Java, refer to 10.2.1, “WebSEAL junctions to SAP NetWeaver AS Java” on page 230.

Note: When creating the junction, ensure that the junction name matches the value used in the redirect.html file. Using the sample listing provided above, the junction would be called /sapasjava.

The junction to the SAP NetWeaver Application Server ABAP does not require any particular settings, apart from those required to connect to the web interface. The Tivoli Access Manager WebSEAL configuration options (configured using webseald-instance.conf) will be dependent on the application being accessed on the SAP NetWeaver Application Server ABAP server.

For more information in regards to creating the junction from Tivoli Access Manager and SAP NetWeaver Application Server ABAP, refer to 10.1.1, “WebSEAL junctions to SAP NetWeaver AS ABAP” on page 224.
10.5.5 Configuring SAP NetWeaver AS Java

This section provides the information required to configure SAP NetWeaver Application Server Java to accept the HTTP header passed from Tivoli Access Manager WebSEAL containing the authenticated user to generate the SAP logon ticket.

Deploying a single sign-on ticket-generating application
For SAP NetWeaver Application Server Java to generate the SAP logon ticket, an authenticated user must access a protected application deployed on SAP NetWeaver Application Server Java. This can be done using an existing protected application by referencing one of its images. The application requires allowing the role everyone access to the referenced image.

Alternatively, a simple application has been provided with the following IBM DeveloperWorks article:

This link allows you to download the SSOTicket.zip file. It contains the SSOTicket.ear file, which can be used with the redirection page that we created earlier in the above sections.

The simple application contains one image, named 1x1.gif, that is accessed by the redirection page. Access to this image is restricted to the everyone group. This requires that the user is authenticated by SAP NetWeaver Application Server Java, allowing us to make use of the SAP logon ticket.

To deploy the sample application, perform the following steps in the SAP Visual Administrator:

1. Expand server - Deploy.
2. Click Deploy & Start.
3. In the File edit control, enter the full location and name of the simple application, SSOTicket.ear. Click OK.
4. If you receive a warning message about the deploy location, click OK.
5. In the Deploy dialog, click OK. The simple application will now be deployed to SAP NetWeaver Application Server Java.
6. To confirm a successful deployment, change the Deployed Components view to **Application**.

7. Within the list of deployed components, locate the entry for `sap.com/SSOTicketEar`. The application should be started. If not, click **Start Application**.

After the application is deployed, attempt to access the image using a browser. The image is accessed via this URL:

http://as-java:port/SSOTicket/1x1.gif

You should receive a prompt for authentication and a blank page result after successful authentication.

**Configuring single sign-on to the sample ticket generating application**

To configure single sign-on to the sample application follow the steps outlined in 10.2.1, “WebSEAL junctions to SAP NetWeaver AS Java” on page 230.

Follow each of the steps outlined in this section to configure single sign-on between Tivoli Access Manager WebSEAL and the sample application deployed above.

This is a summary of the integration procedure:

1. Tivoli Access Manager WebSEAL is configured with a junction to SAP NetWeaver Application Server Java. The junction supplies the Tivoli Access Manager authenticated user ID to SAP NetWeaver Application Server Java in an HTTP header.

2. SAP NetWeaver Application Server Java is configured with a J2EE Login Module to read the HTTP header and validates the user ID using the SAP NetWeaver Application Server Java User Management Engine.

3. After successful validation, the browser is supplied with the MYSAPSSO2 cookie, that is, the SAP logon ticket.

To confirm that the integration is working correctly, examine the list of cookies to ensure that the MYSAPSSO2 cookie is supplied when testing the integration.

**10.5.6 Configuring SAP NetWeaver AS ABAP**

This section provides information about how to configure SAP NetWeaver Application Server ABAP to accept SAP logon tickets generated by SAP NetWeaver Application Server Java.
Configuring trust with SAP NetWeaver AS Java

SAP NetWeaver Application Server ABAP must be configured to trust the SAP logon ticket generated by SAP NetWeaver Application Server Java. This is done using the STRUSTSSO2 transaction by loading the SAP NetWeaver Application Server Java server certificate into the certificate list and ACL of the SAP NetWeaver Application Server ABAP.

For more information about completing this refer to the SAP Library at the following URL:


10.5.7 Testing the configuration

To test for a successful configuration, access an SAP NetWeaver Application Server ABAP application via Tivoli Access Manager WebSEAL. After authenticating to Tivoli Access Manager WebSEAL, the browser should display the SAP NetWeaver Application Server ABAP application without requiring further authentication.

For example, open the browser and navigate to the SAP GUI for HTML (WebGUI) application:

http://webseal_fqdn/sapasabap/sap/bc/gui/sap/its/webgui/

10.6 Conclusion

In this chapter we illustrated the integrations provided by IBM between Tivoli Access Manager WebSEAL and the SAP NetWeaver Application Server ABAP. This chapter covered the integration between Tivoli Access Manager WebSEAL and SAP NetWeaver Application Server Java. It also provided information about a hybrid integration, which allows customers with both SAP NetWeaver Application Server ABAP and SAP NetWeaver Application Server Java to integrate across both platforms by consuming the SAP login ticket.

Each of the sections gave the reader the ability to understand how IBM Tivoli Access Manager WebSEAL and SAP platforms can work together to provide additional protection by ensuring that authentication and authorization are managed at a central point.

Chapter 12, “Access management use cases” on page 283, provides additional information by illustrating different user cases, best practices, and additional
scenarios when considering the integration between Tivoli Access Manager WebSEAL and the SAP NetWeaver platforms.
IBM Tivoli Federated Identity Manager

This chapter provides an overview of the integrations between IBM Tivoli Federated Identity Manager and SAP NetWeaver Application Server-based systems. The following items are covered:

- “Integration with SAP NetWeaver AS Java” on page 260
- “Security Token Service trust module for SAP login ticket” on page 272

Additional use case scenarios that include the usage of IBM Tivoli Federated Identity Manager are described in Chapter 12, “Access management use cases” on page 283.
11.1 Integration with SAP NetWeaver AS Java

SAP NetWeaver Application Server (both Java and ABAP) provides the ability to use the SAML protocol to sign on to its applications (for example, SAP NetWeaver Portal). IBM Tivoli Federated Identity Manager can be used as an assertion source (identity provider) for federated single sign-on to SAP NetWeaver Application Server-based applications.

This chapter describes how to configure single sign-on for SAP NetWeaver AS Java with Tivoli Federated Identity Manager using the SAML protocol and the Browser/Artifact profile.

11.1.1 Introduction

Security Association Markup Language (SAML) is a standard produced by the Security Services Technical Committee (SSTC) within the Oasis Standards Organization. SAML consists of two distinct pieces of functionality:

- The SAML assertion (used to transfer information about a user)
- The SAML protocol (the means of exchanging an SAML assertion)

Full details on SAML are available here:

http://www.oasis-open.org/committees/security

SAML 1.0/1.1 and 2.0 (both ratified as standards) define push-based protocols, meaning that the SSO request is initiated from the identity provider and pushed to the service provider. SAML provides for these:

- Browser/POST profile
- Browser/Artifact profile

The difference between these is how the actual security information (vouch for token) is exchanged between an identity provider (in this case Tivoli Federated Identity Manager) and service provider (SAP AS-Java):

- With a Browser/POST profile, an SAML assertion (vouch or token) is included in the response that is sent to the service provider as part of an HTML form. This is a front channel exchange of the SAML assertion.

- With a Browser/Artifact profile, a pointer to the SAML assertion (called an artifact) is included in the query_string of an HTTP 302 redirect to the service provider. The service provider in turn issues a direct SOAP/HTTP request back.
SAP NetWeaver Application Server Java supports the Browser/Artifact profile (Figure 11-1).

This is the process for when a user accesses an SAP AS-Java application configured for an SAML authentication through Tivoli Federated Identity Manager:

1. The user navigates to an SAP AS-Java application via Tivoli Federated Identity Manager, causing an SSO trigger. The application’s target URL is contained in the request.
2. The user is required to authenticate to Tivoli Federated Identity Manager, if not already authenticated.
3. Tivoli Federated Identity Manager generates an assertion for the request.
4. Tivoli Federated Identity Manager links the assertion to an artifact in its assertion cache.
5. Tivoli Federated Identity Manager returns the artifact to the browser (contained in a query string parameter along with the original target URL) and instructs the browser to redirect to the application.

6. Optionally, a generic SAP SAML receiver servlet is used as the SAML receiver instead of the application. The generic SAML receiver simply redirects the browser to the application with the artifact. This allows for Tivoli Federated Identity Manager to provide SAML SSO to any SAP AS-Java application configured for SAML on the one SAP AS-Java environment. This integration describes a configuration using the generic SAML receiver.

7. The application is configured with the SAP SAML Login Module. The SAML Login Module evaluates the artifact by validating the response. The source site (Tivoli Federated Identity Manager) is determined using information contained within the artifact.

8. The SAML Login Module generates an assertion request (using SOAP) and sends it to Tivoli Federated Identity Manager. This request contains the artifact originally generated by Tivoli Federated Identity Manager.

9. Tivoli Federated Identity Manager performs a lookup in its assertion cache for the appropriate assertion using the artifact as a lookup key.

10. The assertion response is returned from Tivoli Federated Identity Manager to the SAML Login Module.

11. The SAML Login Module analyzes the assertion and authenticates the user.

12. Assuming a successful authentication, the resource is returned to the user.

See the integration package and the integration guide for more detailed information: More detailed information about IBM Tivoli Federated Identity Manager and the integration with SAP solutions can be found at the IBM Support Portal.

The integration package also provides sample mapping rules for SAML 1.0 and SAML 1.1.

Find the Tivoli Federated Identity Manager SAP NetWeaver Application Server Java Integration Adapter at the following location:

11.1.2 Integration process

This section details the steps required to achieve this integration. It contains the following steps:

- Tivoli Federated Identity Manager configuration
- SAP NetWeaver Application Server Java configuration

**Tivoli Federated Identity Manager configuration**
The following sections describe the integration steps required on Federated Identity Manager.

**Creating a federation**
Perform these tasks to create a new federation that uses the SAML SSO protocol.

1. Open the Integrated Solutions Console.
2. Select **Tivoli Federated Identity Manager → Federation Management → Federation**.
3. Click **Create**. The General Information page displays.
4. Enter a name for the federation, such as **SAP AS-Java**.
5. Select **Identity Provider** as your role. Click **Next**. The Contact Information page is displayed.
6. Enter the appropriate company and contact information. Click **Next**. The Federated Protocol Management page is displayed.
7. Select **SAML 1.0**. Click **Next**. The Point of Contact Server page is displayed.
8. Enter the appropriate Point of Contact (this is, the machine with the Tivoli Federated Identity Manager SPS installed). Throughout this chapter, the value `tfim.myexployer.com` will be used as an example. Using this example, enter this value:

   `https://tfim.myexployer.com/FIM`

   Click **Next**. The Signatures page is displayed.
9. Do **not** select **Sign SAML messages for Artifact profile**.
10. Select the appropriate keystore and enter the keystore password.
11. Click **List Keys**.
12. Select the appropriate key to use as the signature. Click **Next**. The SAML Message Settings page is displayed.
13. Accept the default values. Record the value of the **Attribute Resolution Service URL**. Click **Next**. The Configure Security Token page is displayed.
14. Accept the default values, or enter the appropriate values, taking clock skew into account. The SAML Data page displays. Click Next. The Identity Mapping page displays. This page defines how a Tivoli Federated Identity Manager Options identity is mapped to an SAP AS-Java identity.

15. How an identity is mapped depends on your specific requirements and is outside the scope of this section. However, Example 11-1 provides a simple SAML identity mapping rule that maps all Tivoli Federated Identity Manager users to the Administrator SAP user. To use the sample, select Use XSL Transformation for Identity Mapping.

**Attention:** A mapping to use the Administrator user should only be used for testing purposes, and not in a production environment.

16. Click Next. The Identity Mapping page is displayed. Click Browse, locate the sample mapping file in the integration package, and enter the name of the file in XSLT File Containing Identity Mapping Rule.

17. Click Import File. The contents of the file display.

18. Click Next. The Summary page displays.

19. From the SAML Message Settings section, record the value of the following items:
   - Source ID
   - Artifact Resolution Service URL

20. Click Finish. Do not restart WebSphere at this time.

21. Take one of the following actions:
   - Click Done.
   - Click Add partner to immediately move onto the next step.

**Example 11-1 Sample SAML identity mapping**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser" version="1.0">
    <xsl:strip-space elements="*" />
    <xsl:output method="xml" version="1.0" encoding="utf-8" indent="yes" />
</xsl:stylesheet>
<!--
Initially we start with a copy of the document.
-->  
<xsl:template match="@* | node()">  
  <xsl:copy>  
    <xsl:apply-templates select="@* | node()" />  
  </xsl:copy>  
</xsl:template>```
After creating a federation as described above, proceed to the following section.

**Creating a partner**

Perform the following steps to create a new partner for the SAP AS-Java federation. Ignore the first four tasks if *Add partner* was selected in the previous section.

1. Using the Integrated Solutions Console (ISC), select **Tivoli Federated Identity Manager - Federation Management → Partners**.
2. Click **Create**. The Select Federation page displays.
3. Select the federation created in step 1 (for example, SAP). Click **Next**. The Metadata Options page is displayed.
4. This document assumes that you will enter the options manually. Select **Enter SAML settings manually**. Click **Next**. The Contact Information page displays.

5. Enter the appropriate company and contact information. Click **Next**. The SAML Message Settings page displays.

6. Enter the provider ID. For a default installation of SAP AS-Java, this value should be **https://sapserver:50001**, where *sapserver* is the name of the SAP AS-Java machine and **50001** is the SSL port on which the SAP AS-Java instance is listening.

   **Note:** For testing purposes, a TCP connection can be used instead of an SSL connection. To do so, replace *https* with *http* and use the appropriate TCP port (for example, in a default installation, port 50000).

7. Enter the Assertion Consumer Service URL. For a default installation of SAP AS-Java, this value should be **https://sapserver:50001/saml/receiver**. (For testing purposes, a TCP connection can be used instead of an SSL connection.)

8. Click **Next**. The Signature page is displayed. Ensure that *Partner uses Browser POST profile for Single Sign-On* is not selected. Do not validate signatures. Click **Next**. The Configure Security Token page displays.

9. Deselect the *Sign SAML Assertions* option. Click **Next**. The Identity Mapping Options page displays.

10. Click **Next** (this integration uses the identity mapping created for the federation). The Summary page displays.

11. Review the settings and click **Finish** when complete.

12. Click **Enable Partner**.

After creating a partner as described above, proceed to the following section.

**Restarting WebSphere**

In the ISC, click **Restart WebSphere** to enable the changes to the Tivoli Federated Identity Manager configuration.

After restarting WebSphere, proceed to the following section.
**Configuring WebSEAL**

WebSEAL must be configured to become aware of the new federation. See the following URL for details about how to configure WebSEAL appropriately:


Make sure that you complete steps 6 - 9 from the list of instructions.

Ensure that you record the endpoints after configuring WebSEAL.

After restarting WebSEAL, proceed to the following section.

**SAP NetWeaver Application Server Java configuration**

This section covers the steps required to configure SAP AS-Java to support SAML assertions.

**Changing the startup mode for the SAML Service**

Perform the following steps using the Configuration Adapter in the Visual Administrator to change the startup mode for the SAP SAML Service:

1. Select **Server → Services → Configuration Adapter.**
2. Expand **Configurations → cluster_data → server → cfg → services.**
3. Switch to edit mode. Click **Yes.**
4. Select **Propertiesheet tc~sec~saml~service-runtime** and click the pencil representing *Show the details of the selected node.* The Change Configuration page displays.
5. Select **start-up mode.** The Change property entry page displays.
6. In the Custom field, enter the value **always** and click **Apply custom.** This returns you to the Change Configuration page.
7. Click **OK.**
8. Restart the J2EE Engine server process.

After changing the startup mode for the SAML service, proceed to the following section.
Creating a destination
Perform the following steps using Destinations in the Visual Administrator to create a new destination. The destination defines the parameters to connect the Tivoli Federated Identity Manager federation:

1. Select Server → Services → Destinations.
2. Expand Destinations → HTTP.
3. Click New and enter a name for the new destination (for example, TFIM). Click OK.
4. Enter the URL of the Tivoli Federated Identity Manager SOAP Endpoint. The value can be retrieved by viewing the properties of the federation in the ISC. Using the example values above, the entry would contain this:
   https://tfim.myemployerx.com/TFIM/sps/SAP_AS-Java/saml/soap
5. Leave blank the values for System ID, Client, and Language.
6. Select the appropriate authentication mechanism. The mechanism selected will depend on your deployment scenario. For testing purposes, assuming that WebSEAL is used as the authentication server, the sec_master credentials could be used with BASIC authentication.
7. Click Save.

Configuring the SAML parameters
Perform the following steps using the Configuration Adapter in the Visual Administrator to change the SAML parameters:

1. Select Server → Services → Configuration Adapter.
2. Expand Configurations → saml → Configuration.
3. Switch to Edit mode. Click Yes.
4. Select PartnersInbound and click Create a node below the selected node. The Create page displays.
5. Enter a name for the partner (for example, TFIM).
6. Click Create. A new node is created.
7. Click Close Window.
8. Expand the new node.
9. Double-click DestinationName. Enter the name of the destination created in step 3 (for example, TFIM).
10. Double-click **SourceID**. Enter the value of the Tivoli Federated Identity Manager federation Source ID prefixed with `B64:`. This is the source ID that you recorded when configuring the identity provider federation on the Tivoli Federated Identity Manager system.

11. For testing purposes, the `PermitInsecureConnections` parameter (located under Configurations → saml → Configuration → Settings) can be set to `true`. In a production environment, this value should be set to `false`.

After configuring the SAML parameters, proceed to the following section.

**Adjusting the login module stack for using SAML**

Perform the following steps using the Security Provider service in the Visual Administrator to adjust the login modules that apply to the application that is to be configured for SAML assertions. These steps are to be performed for each template or application that is to support SAML assertions, for example, the basic template:

1. Select **Server → Services → Security Provider**.
2. Select the **Authentication** tab.
3. Switch to **Edit** mode.
4. Select the appropriate authentication template, such as `basic` (or `ticket` for SAP NetWeaver Portal. See the note below).
5. Click **Add New**. The Available Login Modules page displays.
6. Select **SAMLLoginModule**. Click **OK**. The SAML Login Module is added to the end of the list of login modules.
7. Select **SAMLLoginModule** and click **Modify**. The Edit Login Module page displays.
8. Set Position to 1.
9. Ensure that Flag is set to SUFFICIENT. Click **OK**.
11.1.3 Testing the integration

Perform the following tasks to test the configuration of SSO using SAML assertions:

Note: This test performs sign-on to the SAP NetWeaver Portal. If SAP NetWeaver Portal is not available, the User Administration servlet (/useradmin/userAdminServlet) can be used along with the Basic authentication template.

1. Adjust the Login Module Stack for the ticket template, as outlined in “Adjusting the login module stack for using SAML” on page 269.
2. Navigate to the SSO trigger. Using the example in this guide, the SSO trigger would be this:

3. Authentication to Tivoli Federated Identity Manager is required because the user has not yet authenticated.

The Portal welcome page should be displayed for the Administrator SAP user.

The integration is now complete.

11.1.4 Removing the integration

The uninstall process is a reverse of the installation process described above.

11.1.5 Troubleshooting

Let us look at a particular troubleshooting situation here.

**Symptom**
After authenticating to Federated Identity Manager, you are redirected to the SAP application, but you are prompted to log in to it.

**Cause**
SAML authentication has failed.

**Solution**
Review the security log using the SAP Log Viewer (refer to SAP documentation for its location).

Look for SAML-related messages with **Severity of Error**.

If all the steps outlined in this guide are followed, the most common cause of SAML authentication failure is clock skew between the Federated Identity Manager server and the SAP AS-Java server.

**Note:** For testing purposes, a TCP connection to the SAP application can be used instead of an SSL connection. To do so, replace the TARGET parameter to specify `http` instead of `https`. 
11.2 Security Token Service trust module for SAP login ticket

Tivoli Federated Identity Manager provides federated identity management between organizations. The Security Token Service (STS) component of Tivoli Federated Identity Manager uses module chains to perform identity validation, mapping, and issuance.

The SAP application provides an identity in the form of the SAP login ticket. This logon ticket is a Base64-encoded representation of an SAP user's identity. IBM provides a trust module for Tivoli Federated Identity Manager that integrates with this login ticket. This technique allows Tivoli Federated Identity Manager to consume, validate, and issue identity tokens using this logon ticket.

An important note is that the logon ticket is owned by the SAP application, so Tivoli Federated Identity Manager is unable to issue a representation of this.

Practical applications of this module are cases where Tivoli Federated Identity Manager is required to validate an incoming SAP identity to map it to a different form of identity.

11.2.1 Tivoli Federated Identity Manager STS modules

These modules run in the Tivoli Federated Identity Manager STS to enable integration with other identity systems.

The Tivoli Federated Identity Manager STS is a WS-Trust compliant service that allows users to validate, exchange, and issue tokens. Within the service is a set of module chains called trust chains. It is possible to have many trust chains within the service, so to determine which chain to invoke it looks at the AppliesTo and Issuer elements (amongst others) in the request.

Each trust chain consists of one or more modules that can either validate, map, issue, or exchange tokens. The standard chain has the following sequence:

1. Validate.
3. Issue (or exchange).

The validate and issue modules tend to be modules that are token specific, such as UsernameToken, X509, or SAML. Traditionally, the map module is a module that contains an XSL stylesheet, though customized mapping modules can be implemented in other languages, such as Java.
11.2.2 SAP module

This identity token module enables the STS to validate identity tokens containing SAP login tickets.

This integration provides a trust module for Tivoli Federated Identity Manager that can be used in a trust chain to validate an SAP user identity issued in an SAP Login Ticket by an SAP system. The SAPSSOEXT library is used to access information.

Figure 11-2 provides an overview of the validation process when used in an STS trust chain.

Figure 11-2  STS trust chain validation process
11.2.3 Integration process

Let us take a closer look at the integration process.

Prerequisites
A number of prerequisite actions are required before attempting to use the STS module. Follow the steps outlined in the following sections to perform these actions.

Downloading the SAP Cryptographic Software
The SAP Cryptographic Software, provided by SAP, is required to validate the signature of the SAP Login Ticket issuing system. To download the library, perform the following actions:

1. Navigate to the SAP Software Downloads site:
   http://service.sap.com/download
   You need a valid SAP user ID to log in.

2. Select Download → SAP Cryptographic Software.

3. Download the appropriate SAP Cryptographic Library for your platform.

4. Extract the contents using SAPCAR (also available from SAP). The extraction should contain a file called ticket and a subdirectory for your platform. Inside the subdirectory, locate the sapcrypto.dll file (Windows) or libsapcrypto.so (UNIX).

5. On the Tivoli Federated Identity Manager machine, create a new directory for the SAP security files. For example, on Windows: C:\sec.

6. Copy sapcrypto and ticket to the new directory.

See the integration package and the integration guide for more detailed information: The integration package provides the module, the configuration file, a sample STS request, and the documentation to enable the integration between Tivoli Federated Identity Manager and the SAP Login Ticket. The module is designed to run in a module chain in the Tivoli Federated Identity Manager Security Token Service (STS) in validate mode.

Find the Tivoli Federated Identity Manager SAP Login Ticket Integration Adapter here:
Downloading the SAPSSOEXT library

The SAPSSOEXT library is the core to performing the login ticket validation. The module wraps this library within the STS module implementation. To download the library, perform the following actions:

1. Navigate to the SAP Service Marketplace:

   http://service.sap.com/patches

2. Click the **Download** tab, then navigate to SAP Support Packages and Patches → Entry by Application Group → Additional Components → SAPSSOEXT → SAPSSOEXT then platform.

3. Download *SAP SSO EXT lib for SAP logon ticket verification*.

4. Uncompress the SAR file using **SAPCAR** (available from SAP).

5. Copy *sapssoext.dll* (Windows) or *libsapssoext.so* (UNIX) to a Tivoli Federated Identity Manager accessible path. For example, on Windows:

   C:\Windows\System32

Obtaining the issuing system’s certificate

The SAP system that issues the SAP Login Ticket (token) signs the token supplied to the STS module. The SAPSSOEXT library requires access to the issuing system’s certificate as part of the validation process. The method of obtaining the issuing system’s public certificate is dependent on the SAP system that is issuing the ticket. The method outlined below is suitable for an SAP Portal 6.0 ticket-issuing system.

Perform the following steps to obtain the issuing system’s certificate:

1. As an administrative user, navigate to your SAP Portal server.

2. Select the **System Administration** tab, then **System Configuration** → **Keystore Administration**.

3. Ensure that SAPLogonTicketKeypair-cert is selected.

4. Click **Download verify.pse File**.

5. Save the file to an appropriate location.

After these prerequisite actions are complete, you can now proceed to deploying the STS module.

Deploying

The following sections describe the process of deploying the STS module.
**Deploying the ISC message labels**
The SAP Token STS Module contains the classes required to display the module's GUI in the IBM Solution Console (ISC). The module's JAR file, `itfim-sts-saptoken.jar`, must be copied to the appropriate location in your Tivoli Federated Identity Manager installation, as follows:

1. Locate the ISC WEB-INF\lib directory. The location is dependent on your installation environment. In a Tivoli Federated Identity Manager installation using embedded WebSphere, this directory is located here:
   
   `FIM_install\ewas\systemApps\isclite.ear\itfim-fimconsole-e.war\WEB-INF\lib`

2. Copy `itfim-sts-saptoken.jar` to the above directory.

3. Restart WebSphere.

**Deploying the SAP Token STS Module**
The SAP Token STS Module must be deployed to the Tivoli Federated Identity Manager run time as follows:

1. Create a new directory in `FIM_install\plugins` and name it `com.tivoli.am.fim.sts.saptoken_1.0`.

2. From the integration package, copy `com.tivoli.am.fim.sts.saptoken_1.0/itfim-sts-saptoken.jar` and `com.tivoli.am.fim.sts.saptoken_1.0/module.xml` to the new directory.

3. In the `FIM_install\FIM\pkg` directory, open `software.properties`.

4. Increment the value of `com.tivoli.am.fim.rte.software.serialId`.

5. Save and close `software.properties`.

6. In the ISC, navigate to **TFIM → Domain Management → Runtime Node Management**.

7. A message states that a new run time is available. Click **Deploy Runtime**.

8. When the deployment of the run time is complete, restart WebSphere.

**Configuring the STS Module**
After it is deployed, a module type and module instance must be created for the SAP Token STS Module, as described in the following sections.

**Creating a new module type**
To do this, follow these steps:

1. Navigate to **TFIM → Configure Trust Service → Module Types**.

2. Click **Create**.
3. Enter the module name:
   com.tivoli.am.fim.sts.saptoken
4. Enter the module version number:
   1.0
5. Enter the exposed class ID:
   SAPTokenSTSModule
6. Click Finish.

Creating a new Module Instance
To do this:
1. Click Restart WebSphere.
2. Navigate to TFIM → Configure Trust Service → Module Instances.
3. Click Create.
4. Select SAPTokenSTSModule. Click Next.
5. Enter an appropriate module instance name. For example:
   SAP Token STS Module Instance
6. Enter an appropriate module instance description. For example:
   SAP Token STS Module Instance
7. Click Finish.
8. Click Restart WebSphere.

11.2.4 Testing

The following sections describe testing procedures to verify that the SAP Module has been successfully deployed and configured.

Creating a trust chain
To test the new module instance, create a trust service chain that includes the new module in validate mode. To do so, perform the following steps:
1. Navigate to TFIM → Configure Trust Service → Trust Service Chain.
2. Click Create. Click Next.
3. Enter a chain name, for example, SAPToken.
4. Enter a description.
5. Select a request type of Validate.
6. In Lookup Type, take these steps:
   a. Ensure that Use Traditional WS-Trust Elements (AppliesTo, Issuer, and TokenType) is selected.
   b. In the Applies To field, enter this address:
      http://appliesto/sap
   c. In the Issuer field, enter this address:
      http://issuer/sap
      Click Next.
7. Select the new module instance. For example:
   SAP Token STS Module Instance
8. Select the Validate Mode.
9. Click Add Selected Module Instance to Chain. Click Next.
10. Enter the full path to the SAP Security Library. For example, (on Windows):
      C:\sec\sapcrypto.dll
11. Enter the full path to the SAP PAB Filename. For example (on Windows):
      C:\sec\verify.pse
12. Enter the SAP PAB Password. By default, the password for verify.pse is blank. Click Next.
13. Click Finish.
14. Click Restart WebSphere.

Obtaining an SAP Login Ticket
The test requires an SAP Login Ticket, supplied by the SAP ticket issuing system.

The best method to obtain a ticket for testing is to authenticate to the issuing system using a browser that provides a method to easily obtain the value of cookies stored in memory. After it is authenticated, locate the cookie named MYSAPSSO2 and copy its value to a text file (for example, ticket.txt).

Sending a request to the STS
The simplest method of sending a request to the STS is to use a command-line utility (for example, cURL) to send a pre-formatted STS Request with a modification of the relevant values (for example, setting the value of the BinarySecurityToken element to that obtained in the previous step).
The integration package contains a pre-formatted STS Request that can be used for this purpose. To use cURL to send the request, take these steps:

1. Copy the `sts-request.xml` file (supplied with the integration package) to a temporary folder.
2. Open the `sts-request.xml` file for modification.
3. Locate the text **INSERT TOKEN HERE** in `sts-request.xml`, and replace it with the value contained within the ticket obtained in the previous section.
4. Ensure that the values for `wsp:AppliesTo` and `wst:Issuer` match the values used when creating the trust chain. Using the examples provided above, the respective values would be `http://appliesto/sap` and `http://issuer/sap`.
5. Save the modified `sts-request.xml` file.
6. If cURL is not available on your system, locate and install an appropriate package.
7. From a command line, execute the following command:

   ```bash
   curl -v --data-binary @sts-request.xml --header "SOAPAction: ''" http://sts_server:sts_port/TrustServer/SecurityTokenService
   ```

   Where `sts_server` is the hostname or FQDN of your STS server, and `sts_port` is the port on which the STS is listening (for example, 9080). The expected output should be similar to Example 11-2.

   **Example 11-2  cURL output example**

   ```xml
   * About to connect() to sts_server port 9080 (#0)
   * Trying sts_server... connected
   * Connected to sts_server (ip) port 9080 (#0)
   > POST /TrustServer/SecurityTokenService HTTP/1.1
   > User-Agent: whatever > Host: sts_server:9080
   > Accept: */* > SOAPAction: ''
   > Content-Length: 1709
   > Content-Type: application/x-www-form-urlencoded
   > Expect: 100-continue
   >
   < HTTP/1.1 100 Continue
   < Content-Length: 0
   < Date: date_time
   < Server: WebSphere Application Server/6.1
   < HTTP/1.1 200 OK < Date: date_time
   < Server: WebSphere Application Server/6.1
   < Content-Type: text/xml; charset=utf-8 < Content-Language: en-AU
   < Content-Length: 670
   ```
11.2.5 Troubleshooting

The following method can be used to troubleshoot any issues with the SAP Token STS Module.

**Enabling WebSphere diagnostic trace**

Enable and configure the WebSphere Diagnostic Trace to log group com.tivoli.am.fim.trustserver.sts.modules.*=all.

Any exceptions thrown by the SAPSSOEXT library are reported.
Refer to SAP documentation for the definition of the error codes. For example:

- standard error= 4, ssf error= 0
  This indicates an expired token.
- standard error= 1281, ssf error= 0
  This indicates an invalid token.

### 11.3 Conclusion

This concludes our discussion of the IBM Tivoli Federated Identity Manager product and its interoperability options with SAP solutions. In the next chapter we investigate sample scenarios and best practices using IBM Security Access Management solutions in conjunction with SAP systems and applications.
Access management use cases

In this chapter we describe common deployment use case scenarios that can be addressed by the IBM Security Access Management solutions for integration with SAP systems and applications. The use cases are driven by preferred deployment scenarios for SAP environments.

Important: The following scenarios are provided as is and are not officially supported by the IBM support chain. If you have any questions about these use the IBM forums to request assistance.

This chapter covers the following use case scenarios:

- “Authenticating an SAP login ticket in Access Manager for e-business WebSEAL using the external authentication C API” on page 285
- “Single sign-on from WebSphere Portal to SAP NetWeaver AS ABAP (SAP ERP) using the Access Manager for e-business Web Server Plug-in” on page 307
- “Single sign-on for SAP with Access Manager and Microsoft Windows using Kerberos and Access Manager for e-business WebSEAL” on page 312
- “SSO to SAP NetWeaver AS ABAP with Federated Identity Manager using SAML 2.0” on page 328
“Service-based single sign-on to SAP backend systems using Federated Identity Manager and SAML” on page 341
“Integrate SAP into SOA by federating the SAP login ticket” on page 343
“Tivoli Access Manager for e-business sample use case scenarios and best practices” on page 362
“Tivoli Access Manager for Enterprise Single Sign-on SAP use cases” on page 377
12.1 Authenticating an SAP login ticket in Access Manager for e-business WebSEAL using the external authentication C API

The implementation of the authentication service extracts and validates the user name in an SAP Login Ticket (an HTTP cookie), then passes the user name to Tivoli Access Manager for e-business WebSEAL to build a credential.

12.1.1 Introduction

With the growth of extensive networks, it is common for organizations to connect many dispersed systems with different security configurations into an intranet or extranet. Network users expect to navigate through the available systems without having to authenticate multiple times. Network administrators are required to configure systems to reduce the number of authentication requests. One typical example of such an environment is where multiple SAP systems are remotely located to each other (managed by separate departments) and are required to provide an uninterrupted user authentication experience.

Tivoli Access Manager for e-business WebSEAL provides a flexible framework with functions that handle authentication operations that can be easily modified, extended, and replaced. Two common options include the External Authentication C API and the External Authentication Interface (EAI).

For a comparison between these mechanisms see Comparison of external authentication mechanisms for Tivoli Access Manager for e-business WebSEAL here:


This section describes how to make use of the External Authentication C API to customize the authentication process of users of such remotely connected SAP systems.

12.1.2 Scenario

This integration considers two scenarios, as outlined below. After considering each scenario, the benefit of implementing an authentication service to customize the Tivoli Access Manager for e-business WebSEAL authentication process that allows for the processing of the SAP Login Ticket will be realized.
Two sample SAP systems are common to both scenarios:

- SAP-Europe
- SAP-Asia

SAP-Europe is protected by Tivoli Access Manager for e-business WebSEAL, whereas SAP-Asia is a stand-alone SAP system without external security policy management. Global users are required to access both systems to perform their duties.
Scenario 1: First accessing SAP-Europe via Access Manager for e-business WebSEAL, and then accessing SAP-Asia

In this scenario, the user first accesses the SAP-Europe portal, which is protected by Tivoli Access Manager for e-business WebSEAL (Figure 12-1). The user then attempts to access the SAP-Asia portal. Trust has been configured between the SAP systems using documented SAP trust mechanisms.

Figure 12-1 Scenario 1: Accessing SAP-Europe via Access Manager for e-business WebSEAL first, then accessing SAP-Asia
This is the process flow illustrated in Figure 12-1 on page 287:

1. The user sends a browser request to the SAP-Europe portal through Tivoli Access Manager for e-business WebSEAL.

2. Tivoli Access Manager for e-business WebSEAL intercepts the request and, if required, prompts the user for authentication credentials, and then authenticates the user against its user registry.

3. Tivoli Access Manager for e-business WebSEAL forwards the request to SAP-Europe, with an HTTP header containing the username of the authenticated user.

4. SAP-Europe validates the supplied user against its user registry, creating an SAP login ticket upon success.

5. SAP-Europe generates the appropriate response and returns it to Tivoli Access Manager for e-business WebSEAL, along with the SAP login ticket (in the form of an HTTP cookie).

6. Tivoli Access Manager for e-business WebSEAL performs filtering in accordance with the junction configuration and returns the response to the browser, including the SAP login ticket.

7. The user then requires content from the SAP-Asia portal, which results in the browser sending another request, this time to the SAP-Asia portal.

8. SAP-Asia validates the SAP login ticket and authenticates the user against its user registry.

9. Upon successful validation, the requested content is returned to the browser.

This scenario describes a solution that does not require the user to present authentication credentials more than one time. However, this is only possible when the user always first authenticates to Tivoli Access Manager for e-business.
WebSEAL that is protecting the SAP-Europe portal. This might not be an acceptable requirement.

This scenario makes use of the IBM-developed and supported integration package available from IBM. See 10.2, “Integration with SAP NetWeaver AS Java” on page 228.
Scenario 2: First accessing SAP-Asia, and then accessing SAP-Europe via WebSEAL

This scenario considers the access order the other way around. That is, the user first authenticates to the SAP-Asia portal, and then requests content from the SAP-Europe portal via Tivoli Access Manager for e-business WebSEAL (Figure 12-2). As per the previous scenario, trust has been configured between the SAP systems using SAP trust mechanisms.

Figure 12-2   Scenario 2: First accessing SAP-Asia, and then accessing SAP-Europe via Access Manager for e-business WebSEAL
This is the process flow illustrated in Figure 12-2 on page 290:

1. The user sends a browser request to the SAP-Asia portal.
2. The SAP-Asia portal accepts the request and, if required, prompts the user for authentication credentials. It then authenticates against its user registry, creating an SAP login ticket upon success.
3. The content is returned to the browser, along with the SAP login ticket (in the form of an HTTP cookie).
4. The user now requires content from the SAP-Europe portal, resulting in the browser sending another request, this time to the SAP-Europe portal. The request is intercepted by Tivoli Access Manager for e-business WebSEAL.
5. Tivoli Access Manager for e-business WebSEAL prompts the user for authentication credentials and authenticates the user against its user registry.
6. Tivoli Access Manager for e-business WebSEAL forwards the request to the SAP-Europe portal.
7. SAP-Europe validates the user against its user registry.
8. Upon successful validation, the content is returned along with the SAP login ticket.
9. The requested content is returned to the browser, including the SAP login ticket.

Unfortunately, even with the inclusion of the IBM integration package, this scenario requires that the user enter authentication credentials two times, once in step 2 and again in step 5. Ideally, the SAP login ticket created by the SAP-Asia portal can be trusted by Tivoli Access Manager for e-business WebSEAL to authenticate the request, therefore not requiring another prompt for authentication credentials.

12.1.3 Solution

The ideal solution addresses both scenarios by combining an integration solution between Tivoli Access Manager for e-business WebSEAL and the SAP-Europe system, which provides the SAP-Europe system with authentication credentials from Tivoli Access Manager for e-business WebSEAL, along with an implementation of a WebSEAL authentication service, to provide Tivoli Access Manager for e-business WebSEAL with authentication credentials from the SAP login ticket issued by SAP-Asia.
The authentication service uses the External Authentication C API to extract the SAP login ticket from the HTTP headers. This is done in conjunction with the Tivoli Access Manager for e-business WebSEAL cookie authentication functionality that allows Tivoli Access Manager for e-business WebSEAL to extract an HTTP cookie from an incoming request and use its contents to create a credential. Additionally, the implementation makes use of the SAP-provided libraries, allowing for the validation details contained in the SAP login ticket.

How to approach authentication: Note that the authentication service can use either the External Authentication C API or the EAI to provide the required functionality. However, SAP provides native C libraries to validate the SAP login ticket. Therefore, it makes sense to use the External Authentication C API. An equally valid solution can be implemented using an EAI approach.
Figure 12-3 illustrates an overview of the solution. The accompanying text for Figure 12-3 describes scenario two. The steps for scenario one do not change.

This is the process flow illustrated in Figure 12-3:

1. The user sends a browser request to the SAP-Asia portal.
2. The SAP-Asia portal accepts the request and, if required, prompts the user for authentication credentials. It then authenticates against its user registry, creating an SAP login ticket upon success.
3. The content is returned to the browser, along with the SAP login ticket (in the form of an HTTP cookie).
4. The user now requires content from the SAP-Europe portal, which results in the browser sending another request, this time to the SAP-Europe portal. The SAP login ticket created by SAP-Asia is included in the request. The request is intercepted by Tivoli Access Manager for e-business WebSEAL.

5. Tivoli Access Manager for e-business WebSEAL determines that the request requires authentication, therefore calling the authentication service to process the request.

6. The SAP login ticket is validated by the authentication service. The user name is extracted and is passed to Tivoli Access Manager for e-business WebSEAL.

7. Tivoli Access Manager for e-business WebSEAL validates that the user exists in its user registry and creates a credential.

8. Now that Tivoli Access Manager for e-business WebSEAL has a valid credential, it forwards the request to the SAP-Europe portal.

9. SAP-Europe validates the user against its user registry.

10. Upon successful validation, the content is returned along with the SAP login ticket.

11. The requested content is returned to the browser, including the SAP login ticket.

12.1.4 Implementation of the authentication service using the External Authentication C API

This section outlines the steps required to implement the authentication service using the External Authentication C API that extracts and validates the SAP login ticket, returning the username contained within it.
A prototype implementation can be downloaded from IBM DeveloperWorks for the following implementations:

- **Prototype implementation (source: authsaptoken_src)**
  
  Download it here:
  

  The sample source does not provide sapssoext.h and ssoload.c files. Both are required to build the binary code. These files are available in the ssosample folder in the SAPSSOEXT library package. Refer to “Downloading the SAP libraries” on page 299.

- **Prototype implementation (binary: authsaptoken_bin)**
  
  Download it here:
  


### Initialization

During the authentication service initialization phase, the SAP-provided libraries for external SSO and security functionality must be loaded and initialized. Ideally, the parameters required for initializing the libraries should be passed to the authentication service `xauthn_initialize()` function.

Example 12-1 provides a sample code snippet illustrating the gathering of required parameters and library initialization.

**Example 12-1   Fragment of the xauthn_initialize() function**

```c
xauthn_status_t xauthn_initialize (int argc,
            const char **argv)
{

    ...

    // Get the configuration data
    while (argc >= 1)
    {
        // Look for the SAP security library
        if (0 == strcmp("-l", *argv))
        {
```

Chapter 12. Access management use cases   295
if (--argc <= 0)
{
    PrintHelp ();
    goto exception;
}
tConfigInfo.pszSecuLib = *++argv;
}

// Look for the public key file
else if (0 == strcmp ("-p", *argv))
{
    if (--argc <= 0)
    {
        PrintHelp ();
        goto exception;
    }
    tConfigInfo.pszSecuLib = *++argv;
}

// Look for the public key file password
else if (0 == strcmp ("-w", *argv))
{
    if (--argc <= 0)
    {
        PrintHelp ();
        goto exception;
    }
    tConfigInfo.pszPubKeyFilePassword = *++argv;
}

// Look for the SAP security extension shared library
else if (0 == strcmp ("-s", *argv))
{
    if (--argc <= 0)
    {
        PrintHelp ();
        goto exception;
    }
    tConfigInfo.pszSSOExtFile = *++argv;
}
argc--; argv++;
Authentication

The authentication service authentication phase is where the identity contained within the SAP login ticket is validated and returned to Tivoli Access Manager for e-business WebSEAL. At the very least, the SAP login ticket username should be returned to Tivoli Access Manager for e-business WebSEAL. Other attributes can be returned to Tivoli Access Manager for e-business WebSEAL in the form of extended attributes, details of which can be found in “Future work: Extending the WebSEAL credential” on page 306.

Example 12-2 provides a code snippet illustrating the authentication phase validating the SAP login ticket and returning the username to Tivoli Access Manager for e-business WebSEAL.

Example 12-2  Fragment of the xauthn_authentication function

```c
xauthn_status_t xauthn_authenticate(xnvlist_t *authnInfo,
                                 xauthn_identity_t *ident)
{
    ...

    // Init the ticket variable
    pszTicket = (char*)malloc(TICKET_BUFFER_LEN);
    if (pszTicket == NULL)
```
{  
    st = XAUTHN_S_OUT_OF_MEMORY;
    goto exception;
}
memset (pszTicket, 0, sizeof(TICKET_BUFFER_LEN));

// Get the value of the cookie
if ( XAUTHN_S_COMPLETE != (st = xnvlist_get(authnInfo,
        DEFAULT_COOKIE_NAME, &pszTicket, &nTicketLength)) )
{
    printf ("Failed to get the value of the cookie.\n");
    goto exception;
}

// Validate the cookie/ticket
if ( 0 != PerformValidation (&tConfigInfo, pszTicket, &tTicketData,
        &tITicketFunctions) )
{
    printf ("PerformValidation function failed with error: \n%s\n",
            GetExceptionBuffer());
    goto exception;
}

...  

name = &ident->prin.data.dn;

// Get the username
if (0 != strlen(tTicketData.pszUserID))
    strcpy(username, tTicketData.pszUserID);
else if (0 != strlen(tTicketData.pszPortalUser))
    strcpy(username, tTicketData.pszPortalUser);
else
{
    printf ("Failed to get the value of the username.\n");
    goto exception;
}

// Set the identities username
*name = (char*)username;
ident->prin.prin_type = principalType;

...
return XAUTHN_S_COMPLETE;
}

### Shutdown

The authentication service **shutdown** phase is the ideal location to shut down the SAP security library.

Example 12-3 provides sample code illustrating the shutdown phase.

*Example 12-3  Fragment of the xauthn_shutdown function*

```c
xauthn_status_t xauthn_shutdown(int argc, const char **argv )
{
    printf("ENTER: xauthn_shutdown()\n");

    // Shutdown the SAP library
    tITicketFunctions.Shutdown();

    // Return complete
    printf("Exit: xauthn_shutdown()\n");
    return XAUTHN_S_COMPLETE;
}
```

### 12.1.5 Configuring the environment

This section outlines the process required to configure the solution environment to allow Tivoli Access Manager for e-business WebSEAL to accept and process the SAP login ticket.

**Downloading the SAP libraries**

As mentioned, a couple of SAP-provided libraries are required to implement the solution.

**SAP Security Library and SAP SSOEXT Library**

Follow these steps:

1. Navigate to the SAP Software Distribution Center:
   
   [https://websmp209.sap-ag.de/swdc](https://websmp209.sap-ag.de/swdc)
   
   You will need a valid SAP account.

2. On the left pane, select **Download** to expand the menu.

3. Select **Support Packages and Patches** to expand the menu.
4. Select **Entry by Application Group**.
5. In the right pane, select **Additional Components**.
6. From the list of components, select **SAPSECULIB → SAPSECULIB5.4**.
7. Select **Download the Library for Your Platform**.
8. Go back to **Additional Components**.
9. From the list of components, select **SAPSSOEXT**.
10. Select **Download the Library for Your Platform**.

**Obtaining the SAP issuing system’s public key**

It is required that the public key of the SAP system that creates and issues the SAP Login Ticket be accessible to the authentication service. The SAP website contains detailed information about how to obtain the public key. The method differs for each SAP system.

If the issuing system is an SAP Portal (as per the scenarios), the following steps can be used to obtain the public key file:

1. In the SAP Portal, choose **System Administration → System Configuration → Keystore Administration**.
2. Choose **Content**.
3. Scroll to the bottom of the screen.
4. Choose to download the **verify.pse** file.

**Enabling HTTP header authentication**

For Tivoli Access Manager for e-business WebSEAL to extract the SAP login ticket in the form of an HTTP cookie, Tivoli Access Manager for e-business WebSEAL is required to be configured for cookie authentication.

**Cookie support:** Cookie authentication is available from Tivoli Access Manager for e-business WebSEAL 6.0 Fixpack 03.

Perform the following actions to configure Tivoli Access Manager for e-business WebSEAL for HTTP header authentication:

1. Open the Tivoli Access Manager for e-business WebSEAL configuration file. For example, with a default installation of Tivoli Access Manager for e-business WebSEAL, the configuration file is located at `WebSEAL_root/etc/webseald-default.conf`.
2. Locate the `[http-headers] stanza`. 
3. Specify the protocols to support in your network environment, for example, http, https, or both.

4. Create a new stanza named [auth-cookies].

5. Create a new parameter within the new stanza called cookie.

6. Set the value of the new parameter to MYSAPSSO.

7. Within the [authentication-mechanisms] stanza, locate the http-request parameter.

8. Set the http-request parameter to the value of the authentication service, ensuring that the parameters are appropriately passed, including:
   - The authentication service binary
   - The SAP Security library
   - The SAP public key file and
   - The SAP SSOEXT library
   For example:
   ```
   C:\Progra~1\Tivoli\PDWebRTE\Supp\authsaptoken.dll &
   -l [C:\Progra~1\Tivoli\PDWebRTE\Supp\sapsecu.dll]
   -p [C:\Progra~1\Tivoli\PDWebRTE\Supp\verify.pse]
   -s [C:\Progra~1\Tivoli\PDWebRTE\Supp\sapssoext.dll]
   ```

9. Save the changes and restart Tivoli Access Manager for e-business WebSEAL.

In summary, the configuration file should contain the settings shown in Example 12-4.

Example 12-4  WebSEAL configuration settings

```plaintext
[authentication-mechanisms]
...
http-request = C:\Progra~1\Tivoli\PDWebRTE\Supp\authsaptoken.dll &
   -l [C:\Progra~1\Tivoli\PDWebRTE\Supp\sapsecu.dll]
   -p [C:\Progra~1\Tivoli\PDWebRTE\Supp\verify.pse]
   -s [C:\Progra~1\Tivoli\PDWebRTE\Supp\sapssoext.dll]

[http-headers]
...
http-headers-auth = both

[auth-cookies]
...
cookie = MYSAPSSO
```
12.1.6 Testing the configuration

The most simple method for testing a successful configuration is to log in to an SAP issuing system, then attempt to log in to Tivoli Access Manager for e-business WebSEAL. The user should not be prompted for authentication to Tivoli Access Manager for e-business WebSEAL.

An alternate method is to obtain an SAP login ticket from the SAP issuing system (by authenticating the SAP issuing system and capturing the MYSAPSSO2 cookie) and use a command-line utility, such as cUrl, to pass the cookie to Tivoli Access Manager for e-business WebSEAL.

Taking the cUrl application as an example, enter the following on the command line:

curl --cookie MYSAPSSO2=cookie_contents http://webseal-host/

Run Tivoli Access Manager for e-business WebSEAL in the foreground to obtain debug information from the authentication service. For example, at the command prompt, type this:

C:}\webseald -foreground

**Successful result**

The expected result upon successful implementation and configuration should be similar to Example 12-5.

*Example 12-5  Sample output of successful authentication*

```
ENTER: xauthn_initialize()
xauthn_initialize() Parameters:
  param[0] = -l
  param[1] = C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapsecu.dll
  param[2] = -p
  param[3] = C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\verify.pse
  param[4] = -s
  param[5] = C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapssoext.dll
SecuLib: C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapsecu.dll
PubKeyFile: C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\verify.pse
PubKeyFilePW: (null)
SSOExtFile: C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapssoext.dll
EXIT: xauthn_initialize()
ENTER: xauthn_authenticate()
ENTER: PerformValidation()
EXIT: PerformValidation()
```

**************************************************
The ticket

AjExMDAgABRwb3J0YWw6QWRtaW5pc3RyYXRvcogAE2Jhc2ljYXVoGVudGljYXRpb24BAAAACAMwMDA
AANKMkUEAwMDTA3MTIxMjIzMjk6FAAQAAAAICgAA%2FwD2MIHzBgkqhkiG9wOBBwKggEuwgeICAQEExC
AJBgUrDgMCAGwUAsGCSqGSIb3DQEHAIEyRMIBATAMAAYJKoZIhvcNAQkDQgUAEwQwMC4CFQDTd
7DbHzIEHcuY8ly201CXUVJzQIVAIJAFzECBsOVgSjbW8NX2QX6wXyo

was successfully validated.
User     :
Ident of ticket issuing system:
Sysid    : J2E
Client   : 000
External ident of user:
PortalUsr: Administrator
Auth     : basicauthentication
Ticket validity in seconds:
Valid (s): 453840
Certificate data of issuing system:
Subject  : CN=J2E
Issuer   : CN=J2E

*****************************************************

username: Administrator

12.1.7 Troubleshooting

This section describes the troubleshooting methods for the output of the authentication service prototype. Three example failure cases are described:

► An expired ticket
► A misplaced library file
► An invalid public key file

In most error cases, a problem has occurred in the SAP libraries. In such cases, the error is reported using a combination of two error codes:

► The standard error code
► The SSF error code

The remaining error cases are reported using an appropriate Tivoli Access Manager for e-business WebSEAL error code.
**Expired ticket**

The expected result upon an expired ticket should be similar to Example 12-6.

Solution: Renew your SAP login ticket.

**Example 12-6  Sample output of expired ticket**

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPWWA1125W</td>
<td>The data contained in the HTTP header MYSAPSSO2 failed authentication</td>
</tr>
<tr>
<td>ENTER: xauthn_authenticate()</td>
<td></td>
</tr>
<tr>
<td>ENTER: PerformValidation()</td>
<td></td>
</tr>
<tr>
<td>EXIT: PerformValidation()</td>
<td></td>
</tr>
<tr>
<td>PerformValidation function failed with error: The SAP logon ticket could not be verified.</td>
<td></td>
</tr>
<tr>
<td>The standard error code is 4. The SSF error code is 0.</td>
<td></td>
</tr>
<tr>
<td>2008-12-08-11:32:28.953+10:00I-----</td>
<td></td>
</tr>
<tr>
<td>0x13212073 webseald ERROR ias general e:\am600\src\libivacl\azn_authn.cpp 290</td>
<td></td>
</tr>
<tr>
<td>0x000005fc</td>
<td></td>
</tr>
<tr>
<td>HPDIA0115E</td>
<td>Unknown identity type.</td>
</tr>
<tr>
<td>2008-12-08-11:32:28.953+10:00I-----</td>
<td></td>
</tr>
<tr>
<td>0x38CF0465 webseald WARNING wwa http s:\amweb600\src\pdweb\webseald\authn\framework\authn-getcreds.cpp 373 0x0000005fc</td>
<td></td>
</tr>
<tr>
<td>DPWWA1125W</td>
<td>The data contained in the HTTP header MYSAPSSO2 failed authentication</td>
</tr>
</tbody>
</table>

---

**Misplaced library file**

The expected result upon a misplaced library file should be similar to Example 12-7.

Solution: Examine your WebSEAL configuration file. Make sure that the correct path is supplied in the http-request parameter within the [authentication-mechanisms] stanza.

**Example 12-7  Sample output of misplaced library file**

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPDIA0122E</td>
<td>Unable to open shared library</td>
</tr>
<tr>
<td>C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\authsaptoken.dll: 126.</td>
<td></td>
</tr>
<tr>
<td>2007-12-08-11:29:02.078+10:00I-----</td>
<td></td>
</tr>
<tr>
<td>0x13212066 webseald ERROR ias general e:\am600\src\ivauthn\ivpam.c 594 0x000000c2c</td>
<td></td>
</tr>
<tr>
<td>HPDIA0102E</td>
<td>Unable to open shared library.</td>
</tr>
<tr>
<td>2007-12-08-11:29:02.078+10:00I-----</td>
<td></td>
</tr>
<tr>
<td>0x38CF013B webseald FATAL wwa server s:\amweb600\src\pdweb\webseald\daemon\webseald.cpp 709 0x000000c2c</td>
<td></td>
</tr>
</tbody>
</table>
Invalid public key file

The expected results upon an invalid public key file should be similar to Example 12-8.

Solution: Examine your WebSEAL configuration file. Make sure that the correct path is supplied in the http-request parameter within the [authentication-mechanisms] stanza. Also make sure that the PSE file is obtained from the correct SAP system that you are using.

Example 12-8  Sample output of invalid public key file

ENTER: xauthn_initialize()
xauthn_initialize() Parameters:
param[0] = -l
param[1] = C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapsecu.dll
param[2] = -p
param[3] = C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\verify.pse
param[4] = -s
param[5] = C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapssoext.dll
SecuLib:  C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapsecu.dll
PubKeyFile:  C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\verify.pse
PubKeyFilePW:  (null)
SSOExtFile:  C:\Program Files\Tivoli\PDWebRTE\CreateDll\src\sapssoext.dll
EXIT: xauthn_initialize()
ENTER: xauthn_authenticate()
ENTER: PerformValidation()
EXIT: PerformValidation()
PerformValidation function failed with error:

The SAP logon ticket could not be verified.

The standard error code is 20. The SSF error code is 7.

2007-12-08-11:31:29.375+10:00I-----
0x13212073 webseald ERROR ias general e:\am600\src\libivacl\azn_authn.cpp 290
0x0000047c
HPDIA0115E   Unknown identity type.
2007-12-08-11:31:29.375+10:00I-----
0x38CF0465 webseald WARNING wwa http
s:\amweb600\src\pdweb\webseald\authn\framework\authn-getcreds.cpp 373 0x0000047c
DPWWA1125W   The data contained in the HTTP header MYSAPSSO2 failed authentication
Future work: Extending the WebSEAL credential
This section focuses on providing Tivoli Access Manager for e-business WebSEAL with the username contained in the SAP login ticket. However, a number of other attributes are available in the SAP login ticket that can be used for authorization purposes. The authenticate phase of the authentication service can be extended to include code that adds the additional attributes to the Tivoli Access Manager for e-business WebSEAL credential.

Additional attributes are added to the Tivoli Access Manager for e-business WebSEAL credential using the client identity that is returned to Tivoli Access Manager for e-business WebSEAL (xauthn_identity_t). Specifically, extended attributes are added to the xattr_list_t data structure. For complete details on how to extend xauthn_identity_t using the xattr_list_t data structure, refer to the Tivoli Access Manager for e-business Web Security Developer Reference.

Table 12-1 lists the additional attributes that are available from the SAP login ticket.

Table 12-1  Attributes of the SAP login ticket

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>The user ID used to authenticate to the SAP system (non-portal). This value is blank when the user authenticates to an SAP Portal system.</td>
</tr>
<tr>
<td>Client</td>
<td>The SAP client number of the ticket issuing system to which the user authenticated.</td>
</tr>
<tr>
<td>Sysid</td>
<td>The SAP sysid of the ticket issuing system to which the user authenticated.</td>
</tr>
<tr>
<td>PortalUser</td>
<td>The user name of the SAP portal user registry. Use this when the issuing system is an SAP Portal system.</td>
</tr>
<tr>
<td>AuthScheme</td>
<td>The method of authentication to the SAP issuing system (defined in SAP as an authentication scheme).</td>
</tr>
<tr>
<td>Validity</td>
<td>The time, in seconds, of the ticket validity.</td>
</tr>
<tr>
<td>Certificate</td>
<td>The public certificate of the SAP issuing system. The certificate is Asn1 encoded, and any certificate information can be obtained.</td>
</tr>
</tbody>
</table>

Summary
This section described how to build an implementation of an authentication service using the External Authentication C API that allows Tivoli Access Manager for e-business WebSEAL to authenticate an SAP login ticket, supplied by an SAP system. Moreover, it described how to test and troubleshoot such an
implementation. Finally, this section described how the authentication service can be extended to provide extended attributes that are contained in the SAP login ticket, therefore enhancing the authorization capabilities of Tivoli Access Manager for e-business WebSEAL.

The Tivoli Access Manager for e-business WebSEAL flexible authentication framework allows for environments with multiple authentication mechanisms to provide users with an uninterrupted authentication experience. In an environment with both Tivoli Access Manager for e-business WebSEAL and SAP systems, combining the Tivoli Access Manager for e-business WebSEAL authentication framework with the SAP login ticket validation libraries gives Tivoli Access Manager for e-business WebSEAL the ability to authenticate a user that was previously authenticated by an SAP system. Likewise, the Tivoli Access Manager for e-business WebSEAL ability to provide an SAP system with an authenticated user ID ensures that the user experience remains uninterrupted as the user traverses the company's network.

12.2 Single sign-on from WebSphere Portal to SAP NetWeaver AS ABAP (SAP ERP) using the Access Manager for e-business Web Server Plug-in

This use case describes how to achieve single sign-on (SSO) for seamless access to SAP NetWeaver Application Server ABAP-based back-end systems (for example, SAP ERP) using BAPI calls from within a WebSphere Portlet. Therefore, **seamless** means that the user is not prompted for an additional SAP password during that access.

12.2.1 Deployment considerations

The configurations described within this chapter assume an existing WebSphere Portal with a Tivoli Access Manager for e-business configured as an external security manager. It also assumes that LTPA is used as single sign-on option from IBM Tivoli Access Manager for e-business to IBM WebSphere Portal.
The goal of the integration is to extend the existing single sign-on realm, which consists of the Tivoli Access Manager for e-business WebSEAL and the WebSphere Portal, by a new single sign-on realm between WebSphere Portal and SAP NetWeaver Application Server ABAP. Within this new realm, native portlets (SAP portlets) issue internal BAPI calls against SAP ERP, using the SAP Java Connector (SAP JCo) (Figure 12-4).

Within the existing single sign-on realm, Tivoli Access Manager for e-business WebSEAL acts on a behalf of an authenticated user on the downstream system WebSphere Portal. WebSphere Portal trusts Tivoli Access Manager for e-business WebSEAL that it has authenticated the user before, when being presented with an LTPA token as a trusted user identity. Because of this trust relationship, WebSphere Portal does not require a user secret for authentication purposes.

From a conceptual point of view, there exist different ways of implementing the new single sign-on realm:

- WebSphere Portal can convince SAP ERP that it is acting on behalf of the user.
- WebSphere Portal can conceal that it is the actual caller by adopting the user's identity.

For the latter it needs to apply to SAP ERP one of its authentication methods.

Regardless of the chosen method, the overall goal of the new single sign-on realm is to prevent the user from being prompted for credentials as soon as the SAP portlets are called.

In addition, an important requirement is that a user be accountable for his calls to the SAP Portal. This requirement excludes that an SAP portlet applies the credential of a technical user for login to SAP NetWeaver Application Server ABAP.
12.2.2 Principle of using the SAP logon ticket for single sign-on

The actual design of this integration is based on a combination of existing trust relationships.

The first trust relationship is between IBM Tivoli Access Manager for e-business and IBM WebSphere Portal (WPS). In this relationship Tivoli Access Manager for e-business forwards an LTPA token for an authenticated user to WebSphere Portal. The token contains the identity of a user in an encrypted form. WebSphere Portal trusts this identity and regards the user as being authenticated.

The second trust relationship is between Tivoli Access Manager for e-business and SAP NetWeaver Application Server. In this scenario, Tivoli Access Manager for e-business is configured as a trusted external authentication resource to SAP. This means that when Tivoli Access Manager for e-business presents to SAP ERP the identity of a user, for example, as the value of a defined HTTP header variable, then SAP trusts this authentication information and regards the user as being authenticated.

SAP ERP on its part relays the user identity to SAP NetWeaver Application Server ABAP. Because a trust relationship also exists between these two parties, the user is finally authenticated at SAP ERP, which issues an SAP logon ticket as a proof.

Combining these two trust relationships yields a derived relationship between WebSphere Portal and SAP ERP. In this solution, an SAP portlet within WebSphere Portal requests an SAP logon ticket using the Tivoli Access Manager for e-business before issuing the actual request to SAP ERP.

WebSphere Portal does the request to Tivoli Access Manager for e-business on behalf of the user, whose LTPA token it presents. Tivoli Access Manager for e-business accepts this token, because it has been issued by itself. In the same way, WebSphere Portal does the request to SAP ERP on behalf of the user whose logon ticket it has acquired.

12.2.3 Operational model for SAP logon ticket-based single sign-on

In this section the actual operational model, which is compliant to the derived trust relationship, is introduced. One essential feature of this model is to use the Tivoli Access Manager for e-business Plug-in for Web Servers as an external authentication engine for SAP ERP. Similar to Tivoli Access Manager for e-business WebSEAL, the plug-in acts as a security gateway for its associated Web Server. But unlike WebSEAL, it is not a standalone proxy-server, but operates as part of the same process as the Web Server.
The plug-in shares with Tivoli Access Manager for e-business WebSEAL the user registry (LDAP) and policies and access control information with the Policy Server. Thus, it can accept authentication decisions made by the Tivoli Access Manager for e-business WebSEAL, like an LTPA Token. This additional trust relationship lets the plug-in act as a kind of replicated Tivoli Access Manager for e-business WebSEAL instance.

Within this model, the following steps are executed when a user wants to issue a call against the SAP portlet within WebSphere Portal (Figure 12-5):

1. The user is authenticated by Tivoli Access Manager for e-business WebSEAL, which protects WebSphere Portal against unauthenticated and unauthorized users.

2. As a result of a successful authentication, Tivoli Access Manager for e-business WebSEAL sends an LTPA Token to WebSphere Portal, which contains the user's identity in an encrypted way. WebSphere Portal accepts the LTPA Token as a proof of the user's authenticity.

3. The SAP portlet detects that it has no SAP logon ticket within the user's portal session. Therefore, it calls a URL hosted by the SAP Application Server that returns the SAP logon ticket as a cookie within its response message. Before issuing the URL call, the LTPA Token is inserted as a cookie into the request message.
4. The Tivoli Access Manager for e-business Plug-in understands and accepts the LTPA Token and forwards the authenticated user identity as the value of an HTTP header variable to SAP.

5. As a result, the user gets authenticated and SAP returns an SAP logon ticket as a proof of authenticity.

6. The SAP logon ticket is inserted as a cookie into the URL response message.

7. The portlet stores the SAP logon ticket in the user’s portal session and submits it as an input parameter of the JCo/BAPI call to SAP.

8. SAP ERP authenticates the user based on the presented logon ticket.

### 12.2.4 Solution steps for SAP logon ticket-based single sign-on

The following configurations are required to enable this use case scenario:

- Configuring the Tivoli Access Manager for e-business plug-in for Web Server for single sign-on to SAP ERP
- Design and implementation of redirect operation (portlet to plug in) mapping an LTPA Token to SAP logon ticket
- Administration of the SAP logon ticket within the portlet, administrating the SAP logon ticket within the user’s portal session
- Integrating the use of the SAP logon ticket into the portlet development
  - Design of SAP Java Connector call pattern using the SAP logon ticket
  - Embedding the SAP Java Connector call pattern and the redirect call into the portlet development

**Additional information:** Depending on the SAP platforms used, and if SAP NetWeaver Application Server Java is present in the scenario, there are two alternatives to configure the integration between Tivoli Access Manager for e-business and SAP NetWeaver Application Server ABAP. Details for each of the options are described in 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222, and 10.5, “Single sign-on for SAP NetWeaver AS ABAP with WebSEAL in conjunction with SAP NetWeaver AS Java” on page 243.
12.3 Single sign-on for SAP with Access Manager and Microsoft Windows using Kerberos and Access Manager for e-business WebSEAL

The question is often asked, “How can we configure single sign-on for SAP GUI using IBM Tivoli Access Manager for e-business WebSEAL like we can do for our other SAP applications?” Unfortunately, the immediate answer is usually, “You cannot, because Tivoli Access Manager for e-business WebSEAL cannot provide security for non-web facing applications. SAP GUI is a non-web facing application.” Of course, this is correct. However, by utilizing the Kerberos protocol in Microsoft Windows, SAP GUI can be configured to automatically use desktop credentials with which to sign on. When used in conjunction with Tivoli Access Manager for e-business WebSEAL, and WebSEAL is configured to support Kerberos, and the user is only required to sign on once (that is, to the desktop).

**Kerberos caveats:** The specifics of the Kerberos protocol are not covered.

To learn more about Kerberos in Windows see this article:


Refer to the Access Manager for e-business online documentation for complete details on how to configure Kerberos and single sign-on solutions with WebSEAL:


12.3.1 Introduction

This section describes how to configure Tivoli Access Manager for e-business WebSEAL and SAP applications, including SAP GUI, to leverage the Kerberos protocol in order to provide a single sign-on environment for SAP and other applications. A brief comparison is made to demonstrate the added value of using Tivoli Access Manager for e-business WebSEAL instead of a solution based solely on the SAP single sign-on ticket.

The method to integrate Tivoli Access Manager for e-business WebSEAL with SAP web-facing applications is only briefly covered in this chapter. This provides the complete single sign-on picture and avoids rewriting the integration guides. A link to the location of the relevant integration package is provided.
These are the SAP applications covered in this document:

- SAP GUI
- SAP WebGUI
- SAP Internet Transaction Server (ITS)
- SAP NetWeaver Portal
- SAP ERP (R/3)

All servers, including Tivoli Access Manager for e-business WebSEAL, are assumed to be running on Windows.

**SAP Internet Transaction Server integration:** This use case scenario assumes the option to configure SAP Internet Transaction Server (ITS) for trust and single sign-on with SAP NetWeaver Application Server ABAP and SAP ERP.

Note that IBM has deprecated its officially supported integration guide between IBM Tivoli Access Manager for e-business and SAP ITS. This decision was made because as of SAP NetWeaver 2004, the stand-alone SAP ITS has been integrated into the SAP NetWeaver Application Server as an Internet Communication Framework service.

Nevertheless, this section describes the single sign-on option using SAP Internet Transaction Server for environments where this would be an option.

For all other scenarios with the same scope but without the ability to use SAP ITS, there is the option to achieve single sign-on between Tivoli Access Manager for e-business and the integrated SAP Internet Communication Framework service with a new, officially supported method by using the existing *IBM Tivoli Access Manager for e-business SAP NetWeaver ABAP Integration Guide*. This integration is outlined in 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222.

Alternatively, the latest version of the Tivoli Access Manager for e-business integration with SAP NetWeaver Application Server ABAP can be downloaded from the following URL:


### 12.3.2 Scenario

Consider an intranet environment where a company’s users log on to a Microsoft Windows desktop by authenticating to Active Directory. From their desktops, users access an SAP R/3 instance using the SAP GUI or SAP WebGUI using a
browser via the SAP Internet Transaction Server (ITS). Additionally, Internet Explorer (IE) is used to access the company’s corporate portal hosted on SAP NetWeaver Portal. Finally, a number of web application servers, providing various other mission-critical applications, are available to users via their browsers.

In such a heterogeneous environment, without an access management solution, users are required to authenticate to a number of systems. Specifically, the user is required to log on to his Windows desktop and subsequently to each application, at least five logons. Moreover, the user is required to manage the credentials for each of these systems.

Figure 12-6 shows a basic illustration of the environment.
12.3.3 Solution

The described environment is difficult to manage, is inconvenient for the user, and is potentially insecure, as users write their passwords in insecure locations. Therefore, a solution is required to manage user access to the applications and to assist with the management of the user's identities for each application.

The solution for the environment can be broken into two general areas:

- Access management
  
  Reduce the number of times that a user is required to provide authentication credentials to the applications, that is, through the use of single sign-on techniques.

- Identity management
  
  Reduce the administrative overhead of managing the user's identity for each application's user registry.

This section focuses on the access management part of the solution.
As shown in Figure 12-7, the ideal access management solution requires the user to provide one set of authentication credentials (that is, to Active Directory when logging into Windows). Any subsequent authentication requests are then handled by the client machine by supplying authentication information using Kerberos tickets.

Figure 12-7  Access management solution using IBM Security and Microsoft Windows

When using Kerberos (without discussing the specifics of the Kerberos protocol in detail), on successful authentication to Active Directory the user is supplied with a Kerberos ticket. The Kerberos ticket is then used to authenticate the user to any application that implements the Kerberos protocol.
For the web applications that are not configured for Kerberos, Tivoli Access Manager for e-business WebSEAL is used to establish a trust relationship with the web applications to provide the sign-on information. Subsequently, the web application does not require specific coding to handle the Kerberos protocol. Additionally, the trust relationship enables the single sign-on techniques to be extended to the internet.

Why use a Tivoli Access Manager for e-business WebSEAL single sign-on solution instead of an SAP solution based on the SAP single sign-on ticket? SAP makes use of its single sign-on ticket through the use of the MYSAPSSO cookie to provide single sign-on to SAP systems. Furthermore, the single sign-on ticket can be utilized in a heterogeneous environment by extending the non-SAP applications to read and decode the single sign-on ticket. Therefore, you might ask why we would not use SAP single sign-on tickets. In addition to the obvious benefits provided by Tivoli Access Manager through centralized authentication, access, and audit policy (to name just a few), the answer is simple:

- **Less integration overhead**
  
  A significant amount of overhead is required to establish trust between SAP applications based on the SAP single sign-on ticket. In comparison, the process of establishing a Tivoli Access Manager for e-business WebSEAL single sign-on solution for SAP applications is simple.

- **Less modification to non-SAP applications**
  
  For a heterogeneous environment (that is, an environment containing non-SAP applications), no modification, or very little modification, is required to establish a Tivoli Access Manager for e-business WebSEAL single sign-on solution, whereas a significant development effort is usually required to support the use of the SAP single sign-on ticket. Refer to “Configuring other web servers trust relationship with WebSEAL” on page 327 for more details about non-SAP single sign-on solutions.

- **More suitable for mobile users**
  
  As mentioned above, a Tivoli Access Manager for e-business WebSEAL single sign-on solution is easily extended to provide secure access to internet users. Therefore, for mobile users, the user experience is greatly enhanced when accessing the applications from the internet. By making use of Tivoli Access Manager for e-business WebSEAL’s ability to use both Kerberos authentication alongside other methods of authentication suitable for the internet (for example, Basic Authentication and forms authentication), the user can sign on to Tivoli Access Manager for e-business WebSEAL from the internet using the same login credentials as when signing-on to their desktop (or laptop) from within the intranet.
In summary, a Tivoli Access Manager for e-business WebSEAL solution provides a flexible method of access management without the overhead of an SAP proprietary solution.

**Configuring the desktop for Kerberos authentication**

No action is required to enable Kerberos authentication for the initial user login to their desktop. Figure 12-8 shows a very basic illustration of the Active Directory login process, including provision of the Kerberos ticket.

![Figure 12-8  Active Directory login](image)

**Configuring SAP ERP for Kerberos authentication**

The following steps outline how to configure the SAP ERP Application Server for the Kerberos protocol. Figure 12-9 shows the relevant section of the overall solution that is covered in this section.

![Figure 12-9  Configuring SAP ERP (R/3) for Kerberos authentication](image)

**Step 1: Configuring SAP ERP server into Active Directory domain**

To participate in a Kerberos exchange, the SAP ERP Application Server must be a member of, and have an identity in, the Active Directory domain. After the SAP ERP server is added to the Active Directory domain, the SAP ERP service account can then be registered with the Active Directory domain controller. This enables client applications to obtain a Kerberos ticket from the Active Directory domain controller to access the SAP ERP server.
To create and configure the SAP ERP service account:

1. Create a service account using the Active Directory Users and Computers MMC snap-in.

2. Add the new account to the local Administrators groups on the SAP ERP server. This is done using the Computer Management MMC snap-in.

3. On the SAP ERP server, using the Services Control Panel applet, configure the SAP ERP service to log on using the newly created account. The SAP ERP service is typically named SAPServiceName_ServiceNumber. Refer to SAP documentation to assist with locating the correct service.

This publication assumes that the service account is called `<DOMAIN>\sapr3` (where `<DOMAIN>` is the name of the Windows NetBIOS domain). Therefore, for the remainder of this chapter, the SAP ERP service account's Kerberos principal name will be referred to as sapr3@COMPANY.COM (where COMPANY.COM is the Active Directory domain name) and the SAP ERP service account's SNC name is p:sapr3@COMPANY.COM.

**Step 2: Mapping the Kerberos principal to the Active Directory user**

The SAP client does not request the service principal name (SPN) when requiring access to the SAP ERP server. However, the SPN must be set because the Kerberos Domain Controller will invoke the Kerberos user2user protocol for any account that does not have an SPN. Therefore, the SAP ERP Kerberos principal name must be mapped to the Active Directory user that represents the SAP ERP service account, created in step 1.

This mapping requires the Windows `setspn` utility. The `setspn` utility is not loaded on a Windows system by default. It can be obtained from the Windows Support Tools package on the Windows CDs.

To create an SPN for the SAP ERP service account, log in as a domain administrator and execute the following command:

```
C:\>SETSPN -A SAPService<ServiceName>/SAP R/3 host name
<DOMAIN>\sapr3
```

Where `<ServiceName>` is the name of the SAP ERP service and `<SAP R/3 host name>` is the fully qualified name of the SAP ERP server.

**Step 3: Installing the Kerberos runtime on the SAP R/3 server**

To enable the use of the Kerberos protocol by SAP ERP, the Kerberos runtime must be installed on the SAP ERP server.
To obtain and install the Kerberos client:

1. Download the appropriate version from the SAP Service Marketplace from SAP Note 352295.

2. Extract the code and copy gsskrb5.dll to the %windir%\system32 directory, for example, C:\WINNT\system32.

3. Set the SNC_LIB environment variable to the location of the gsskrb5.dll file.

**Step 4: Configuring SNC to use the Kerberos runtime**

SAP Secure Network Communication (SNC) is required to be enabled and configured to use the Kerberos runtime.

Using the SAP GUI, run transaction RZ10 and set the following parameters:

```
snc/enable =1
snc/gssapi_lib = %windir%\system32\gsskrb5.dll
snc/identity/as = p:sapr3@COMPANY.COM
```

**Step 5: Mapping SAP ERP user accounts to Active Directory users**

For each user requiring access to the SAP ERP using the SAP GUI, the user's Kerberos principal name must map to an SAP username.

To set a user's Kerberos principal name, use the SAP GUI (or another means of running an SAP transaction) to perform the following steps:

1. Run transaction SU01.
2. Enter the SAP username.
3. Select the SNC tab.
4. Enter the user's Kerberos principal name in the following format:
   
   Username@COMPANY.COM

   Note the use of case.

5. Save the changes.

**Step 6: Restarting the SAP Application Server**

Restart the SAP Application Service service to enable the changes.

SAP ERP is now configured to accept Kerberos authentication.
Configuring SAP GUI for Kerberos authentication
The following steps outline how to configure the SAP GUI for Kerberos authentication. Figure 12-10 shows the relevant section of the solution covered in this section.

![Diagram showing user desktop and SAP R/3 connected by Kerberos]

Figure 12-10  Configuring SAP GUI for Kerberos authentication

**Step 1: Installing the Kerberos run time on the Windows Desktop**
The Kerberos runtime is required to be installed on each Windows desktop to enable the use of the Kerberos protocol by SAP GUI.

Take these steps to obtain and install the Kerberos client:

1. Download the appropriate version from the SAP Service Marketplace via SAP Note 352295.
2. Extract the code and copy `gsskrb5.dll` to the `%windir%\system32` directory, for example, `C:\WINNT\system32`.
3. Set the SNC_LIB environment variable to the location above.

**Step 2: Configuring SAP GUI to use SNC**
Modify the properties of the SAP ERP connection in the Advanced Options pane in the initial screen of the SAP GUI. Select **Enable Secure Network Communication** and set the SNC name to the value of the SAP service account, for example, `p:sapr3@COMPANY.COM`.

After restarting the SAP GUI, it is now configured to use Kerberos authentication.
Configuring SAP ITS for Kerberos (SNC) authentication to SAP ERP

When configuring the SAP ITS for single sign-on you are required to configure SNC to the SAP Application Server. The SAP Application Server is expecting the SNC communication based on the Kerberos protocol. Therefore, the ITS must be configured to use the Kerberos protocol (Figure 12-11).

Figure 12-11 Configuring SAP ITS for Kerberos (SNC) authentication to SAP ERP

To configure the SAP ITS to use the Kerberos protocol:

1. Modify the ITS instance configuration file (ITSRegistryWGate.xml) to specify the SNC name of the SAP ERP (referred to as the Application Gate, or Agate, by SAP ITS):
   \[
   \text{SncNameAGate = p:sapr3@COMPANY.COM}
   \]

2. Using transaction SM30, modify the USRACLEXT table to add an entry as follows:
   \[
   * \rightarrow \text{p:sapr3@COMPANY.COM}
   \]
3. Using transaction SM30, run report RSSNCCHK to confirm that the SNC parameters are correct.

4. Open the SNC system access control list (table SNCSYSACL, view VSNCSYSACL, type=E) and perform the following steps:
   a. Enter the AGate's SNC name in the SNC name field. The System ID field is optional.
   b. Activate:
      - Entry for RFC
      - Entry for CPIC
      - Entry for DIAG
      - Entry for certificates
      - Entry for external ID
   c. Save the changes.

The SAP ITS is now configured for SNC communication based on the Kerberos protocol.

**Configuring WebSEAL for Kerberos authentication**

The final system requiring configuration for the Kerberos protocol is the Tivoli Access Manager for e-business WebSEAL server (Figure 12-12). Refer to the Tivoli Access Manager for e-business WebSEAL Administration Guide for complete details on how to configure WebSEAL for Kerberos.

The minimum steps required to configure Tivoli Access Manager for e-business WebSEAL for Kerberos on a Windows systems are outlined below.

**Step 1: Configuring WebSEAL server into Active Directory domain**

As for the SAP ERP server, to participate in a Kerberos exchange, the Tivoli Access Manager for e-business WebSEAL server must be a member of, and have an identity in, the Active Directory domain. Refer to Microsoft
documentation for complete details about how to complete this task. At the very least the following actions are required:

- Using the Active Directory Users and Computers MMC snap-in, add the Tivoli Access Manager for e-business WebSEAL server as a member of the AD domain.
- Using the DNC MMC snap-in, create an appropriate DNS entry for the Tivoli Access Manager for e-business WebSEAL server.

**WebSEAL configuration information:** When Tivoli Access Manager for e-business WebSEAL is installed on a non-Windows platform, the Tivoli Access Manager for e-business WebSEAL server is not required to be a member of the AD domain. Refer to the *IBM Tivoli Access Manager for e-business WebSEAL Administration Guide Version 6.1.1*, SC23-6505-01, for details about how to configure Tivoli Access Manager for e-business WebSEAL for the Kerberos protocol when using a non-Windows platform.

**Step 2: Mapping Kerberos principal to Active Directory user**

Take these steps:

1. If required, install the Windows Support Tools from the Windows install CD.
2. From a command prompt, run this:

   ```
   ktpass -princ HTTP/<webseal.company.com>@<COMAPNY.COM> -mapuser <webseal>$
   ```

   Where `<webseal>` is the hostname of the Tivoli Access Manager for e-business WebSEAL server, and `<COMAPNY.COM>` is the domain name of the AD domain.

**WebSEAL account information:** `<webseal>$` represents the Tivoli Access Manager for e-business WebSEAL service running as the machine's local server account. For multiple instances of WebSEAL on the same machine, or when Tivoli Access Manager for e-business WebSEAL is configured to log in as an AD service account, the Windows style username is required to be specified instead of `<webseal>$` (as is done for the SAP ERP service account).
**Step 3: Enabling Kerberos for WebSEAL**
Modify the Tivoli Access Manager for e-business WebSEAL configuration file to enable SPNEGO as follows:

1. Stop the Tivoli Access Manager for e-business WebSEAL server.
2. Modify the Tivoli Access Manager for e-business WebSEAL configure file to enable SPNEGO over SSL:
   ```
   [spnego]
   spnego-auth = https
   
   [authentication-mechanisms]
   kerberosv5 = Tivoli_Access_Manager_install_dir\bin\stliauthn.dll
   ```

**Step 4: Restarting WebSEAL**
Restart Tivoli Access Manager for e-business WebSEAL to enable the changes to the configuration.

**Step 5: Configuring Internet Explorer**
Internet Explorer must be configured to use the SPNEGO protocol to negotiate authentication mechanisms. Perform these steps in IE to enable Kerberos authentication:

1. From the menu, select **Tools → Internet Options**.
2. Select the **Advanced** tab.
3. Scroll down to the Security section.

Tivoli Access Manager for e-business WebSEAL is now configured for Kerberos authentication.
Configuring SAP ITS trust relationship with WebSEAL
To enable single sign-on to the SAP ITS, Tivoli Access Manager for e-business WebSEAL is configured to send the authenticated user ID to the ITS using an HTTP header. The ITS is then configured to trust the user ID provided by Tivoli Access Manager for e-business WebSEAL and proceeds with normal processing for the authenticated user (Figure 12-13).

Refer to the Integration Guide available on the IBM website for details about how to configure the trust relationship between Tivoli Access Manager for e-business WebSEAL and SAP ITS.

Configuring SAP NetWeaver Portal trust relationship with WebSEAL
To enable single sign-on to the SAP NetWeaver Portal, Tivoli Access Manager for e-business WebSEAL is configured to send the authenticated user ID to the Portal using an HTTP header. The Portal is then configured to trust the user ID provided by Tivoli Access Manager for e-business WebSEAL and proceeds with normal processing for the authenticated user (Figure 12-14).
Refer to the *Integration Guide* available on the IBM website for details about how to configure the trust relationship between WebSEAL and SAP NetWeaver Portal:


**Configuring other web servers trust relationship with WebSEAL**

For most web applications, Tivoli Access Manager for e-business WebSEAL can provide the sign-on credentials for authenticated users. This is done using a variety of mechanisms ranging from supplying an HTTP header containing the authenticated user ID (as is done in the SAP ITS and SAP NetWeaver Portal integrations) to supplying Basic Authentication credentials stored in Tivoli Access Manager for e-business WebSEAL's Global Sign-On lockbox.

![Figure 12-15 Configuring other web application servers trust relationship with Tivoli Access Manager for e-business WebSEAL](image)

Refer to *IBM Tivoli Access Manager for e-business WebSEAL Administration Guide Version 6.1.1*, SC23-6505-01, or the list of integration packages available at the following link location for details on how to configure the trust relationship between WebSEAL and any web applications requiring a single sign-on solution:


**Testing the solution**

The following tasks should be completed to test the access management solution:

1. Log in to the desktop.
2. Open IE and navigate to Tivoli Access Manager for e-business WebSEAL's default page.

   The user should be authenticated to Tivoli Access Manager for e-business WebSEAL without a prompt and without error. This indicates that the user has successfully authenticated to WebSEAL using SPNEGO (Kerberos).

The user should be authenticated to the portal without an SAP login prompt and without error. This indicates that the trust between Tivoli Access Manager for e-business WebSEAL and SAP NetWeaver Portal is established.


The user should be authenticated to ITS without an SAP login prompt and without error. This indicates that the trust between WebSEAL and SAP ITS is established. Additionally, this indicates that SNC is successfully configured between SAP ITS and SAP ERP.

5. Using IE, navigate to any other web application servers via a Tivoli Access Manager for e-business WebSEAL junction.

The user should be authenticated to the application without a prompt and without error. This indicates that the trust between Tivoli Access Manager for e-business WebSEAL and the application is established.

6. Open SAP GUI and select the appropriate SAP ERP server.

The user should be authenticated to the application without a prompt and without error. This indicates that the Kerberos authentication is configured for SAP GUI.

**Solution alternative:** Alternatively to the use of SAP Internet Transaction Server for configuring trust and single sign-on with SAP NetWeaver Application Server ABAP, you can use the integration described in 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222.

Another option is the integration of Tivoli Access Manager for e-business with SAP NetWeaver Application Server ABAP in conjunction with SAP NetWeaver Application Server Java, as described in 10.5, “Single sign-on for SAP NetWeaver AS ABAP with WebSEAL in conjunction with SAP NetWeaver AS Java” on page 243.

### 12.4 SSO to SAP NetWeaver AS ABAP with Federated Identity Manager using SAML 2.0

In this use case we discuss a scenario in which a non-SAP client application will connect to an SAP NetWeaver Application Server ABAP-based system. In this example an IBM Lotus Notes® sidebar plug-in connects to an SAP NetWeaver Gateway Server to receive and update SAP Workflow items. The use of SAP
NetWeaver Gateway in this case introduces a new possibility of REST interaction with SAP workflow items by leveraging one standard SAP NetWeaver Gateway service delivered with the product. As a result, the users should be able to use their intranet ID and password (for example, as used in conjunction with Tivoli Access Manager for e-business) to log on to SAP NetWeaver Gateway server, as they might be not aware of their SAP credentials. See Figure 12-16 on page 330 for an illustration of the involved components and the general process flow.

The integration allows you to access the workflow item in the SAP application, update it, and proceed with the SAP workflow out of an interactive Lotus Notes sidebar plug-in.

In this section we discuss the required configurations to establish single sign-on between IBM Tivoli Federated Identity Manager and SAP NetWeaver Application Server ABAP (AS ABAP) using the Security Assertion Markup Language (SAML) standard. Tivoli Federated Identity Manager will be used as an identity provider for single sign-on using SAML 2.0 artifact binding.

### 12.4.1 SAML protocol specifics

General aspects about SAML are already described in this book in 2.4.4, “Security Assertion Markup Language” on page 49, and 11.1, “Integration with SAP NetWeaver AS Java” on page 260. In addition, the artifact resolution and artifact binding are of interest for this use case scenario.

**Artifact Resolution Protocol**

A SAML message is transmitted from one entity to another either by value or by reference. A reference to a SAML message is called an artifact. The receiver of an artifact resolves the reference by sending a `<samlp:ArtifactResolve>` request directly to the issuer of the artifact, who then responds with the actual message referenced by the artifact.

Suppose that an identity provider sends a `<samlp:ArtifactResolve>` request directly to a service provider (via a back channel). In response, the service provider returns the SAML element referenced by the enclosed artifact. This protocol forms the basis of the HTTP Artifact Binding.²

**HTTP Artifact Binding**

For web browser single sign-on, the HTTP POST Binding is commonly used. Either the service provider or the identity provider (or both) use HTTP POST to transmit a protocol message. An entity’s choice of binding is independent of its

---

¹ In this document and use case example we only cover the SAP and Tivoli Federated Identity Manager parts of the scenario.

partner's choice of binding. For example, the service provider can use HTTP POST while the identity provider uses HTTP Artifact.

The HTTP Artifact Binding uses the Artifact Resolution Protocol and the SAML SOAP Binding (over HTTP) to resolve a SAML message by reference. Suppose that a service provider wants to send a `<samlp:AuthnRequest>` message to an identity provider. Initially, the service provider transmits an artifact to the identity provider via an HTTP redirect.

Next the identity provider sends a `<samlp:ArtifactResolve>` request (such as the ArtifactResolveRequest) directly to the service provider via a back channel.

Finally, the service provider returns a `<samlp:ArtifactResponse>` element containing the referenced `<samlp:AuthnRequest>` message. The flow can go in the other direction as well (that is, the identity provider might issue an artifact).3

12.4.2 Process flow

Figure 12-16 illustrates the general process flow for single sign-on to SAP NetWeaver Application Server ABAP and Tivoli Federated Identity Manager.

![Diagram](Figure 12-16 Process flow for single sign-on to SAP NetWeaver Application Server ABAP with Federated Identity Manager)

---

3 Source: <http://en.wikipedia.org/wiki/SAML_2.0#HTTP_Artifact_Binding>
This is the process flow as per Figure 12-16 on page 330:

1. The user calls the application on the runtime environment.
2. A runtime environment (such as an application server) is connected to an LDAP repository and authenticates the user.
3. Application requests SAP data.
4. Gateway redirects to Tivoli Federated Identity Manager for a SAML 2.0 assertion.
5. Tivoli Federated Identity Manager trusts the runtime environment token for the LDAP user.
6. Tivoli Federated Identity Manager asserts the SAP user and redirects to the SAP Gateway.
7. The SAP Gateway collects SAP data for this user.
8. The SAP Gateway sends data to the application.
9. The user receives the response from the application.

To generate the most usability for the users, the request to SAP NetWeaver Gateway should be enriched by a security token (for example, LTPA) before it is sent to the target SAP Gateway. If the token is sent, it gets checked automatically by the connected Tivoli Federated Identity Manager, which then creates the SAML2 assertion. This use of the security token eliminates any manual user login in this flow.

### 12.4.3 Deployment considerations

The following SAP application prerequisites and SAML support specifics need to be considered before the deployment.

**SAP application prerequisites**

For this scenario the following SAP application prerequisites must be fulfilled:

- SAP CryptoLib installed and active.
- HTTPS configured.
- Create SSL client PSE by using transaction trust manager (STRUST).
- Assign role `SAP_SAML2_CFG_ADM` to ABAP admin user.

**SAP SAML 2.0 support for ABAP stack**

SAP applications support two options for SAML-based single sign-on: front-channel and back-channel single sign-on. Because for SAP NetWeaver Gateway artifact authentication is required, we need to use back-channel communication.
12.4.4 SAML 2.0 process flow for back-channel SSO

Figure 12-17 illustrates and describes service-provider-initiated single sign-on with back-channel communication. In this scenario the SAP NetWeaver Application Server ABAP system is the service provider and IBM Tivoli Federated Identity Manager is the identity provider.

---

Figure 12-17  SAML 2.0 flow for back-channel SSO with Federated Identity Manager and SAP AS-ABAP

---

4 See SAP Help for documentation about SAP NetWeaver Gateway at http://help.sap.com/saphelp_gateway20/helpdata/en/71/376ab03f824ea5bfd0a0e3a307205b/frameset.htm

5 Single sign-on to web services in a ABAP stack is supported from SAP NetWeaver 7.0 EHP2 by using SAML 2.0. For details see SAP Help on NetWeaver 7.0 EhP2 here: http://help.sap.com/saphelp_nw70ehp2/helpdata/en/46/631b92250b4fc1855686b4ce0f2f33/content.htm
The following process flow occurs on back-channel single sign-on with SAML 2.0:

1. A user attempts to access an SAP NetWeaver Application Server ABAP resource protected by SAML 2.0.
2. The SAP AS-ABAP server (as the service provider) redirects the user to an identity provider (Tivoli Federated Identity Manager) and includes a SAML artifact referring to the authentication request.
3. Tivoli Federated Identity Manager (as the identity provider) gets the authentication request from the service provider over a SOAP channel.
4. Tivoli Federated Identity Manager queries the user for authentication credentials.
5. The user or user agent presents the requested credentials.
6. Tivoli Federated Identity Manager returns the user to the AS-ABAP server with a SAML artifact referring to the authentication response.
7. The AS-ABAP server gets the authentication response from Tivoli Federated Identity Manager over a SOAP channel.
8. The user is authenticated to access the requested AS-ABAP resource.

**12.4.5 Configuration**

The configuration of both systems will need several steps. Be sure that you also have file system access to the installation directory of the Tivoli Federated Identity Manager to access exported certificates.

**Tivoli Federated Identity Manager: Creating a domain**

In this section we start with a new installed Tivoli Federated Identity Manager. The domain needs to be created first. The Getting Started page shows a detailed overview about the system status. Take these steps:

1. Click the **Manage Domains** link to start the creation wizard.

   Use the wizard to create a new domain with a local connection as Management Service Endpoint and accept the default WebSphere Target Mapping.

2. Finish the wizard.

   The domain creation was successful, and you get asked whether it should be activated directly and whether the Runtime Node Management should be opened. Accept both by clicking **OK**.

---

3. Load the configuration changes into the Tivoli Federated Identity Manager run time.

4. Deploy the run time by clicking **Deploy Runtime**. This takes a while, but you can always use the link to refresh the current status. Or you can monitor the CPU load to see when it is finished.

5. Select the new run time by using the check box in the front and selecting **Configure**. As a result, the option in the row will get selected.

6. Load the new settings into the Tivoli Federated Identity Manager run time.

7. Set the local WebSphere as **Point of Contact**. Make sure that it uses a free port for the SOAP connection (for example, 8881, as the standard WebSphere Application Server installation already uses 8880).

---

**Tivoli Federated Identity Manager: Creating a federation**

In a federation the communication configuration will be stored. This could then be used with one or more partners. Use again the link on the Getting Started page to start the wizard:

1. On the General Information page enter a federation name and select the role as **Identity Provider**. Click **Next**.

2. Enter company information and click **Next**.

3. On the Federation Protocol page select **SAML 2.0**, as we will use this protocol. The screen will need a moment to refresh. Click **Next**.

4. For **Point of Contact Server** in the first turn the local WebSphere environment will be used. Enter the local hostname and the http port into the field (https can be also used here). Click **Next**.

5. For profile choose **Basic: Web Browser SSO, Single Logout** and click **Next**.

6. For the signature, a **signing key** is needed. For first-time setup and test you can use the default keystore and the standard password `testonly` (which needs to be modified in production environment). Enter it and select **List Keys** to show the available keys. In default there is only one. Select the `fimdemo` certificate and click **Next**.

7. For the encryption certificate do the same as for the signing certificate and click **Next**.

8. For the **SAML Message Settings** and the **SAML Assertion Settings** accept the defaults and click **Next** two times.

9. Select **Use XSL or JavaScript transformation for identity mapping** as the mapping configuration and click **Next**.
10. Select the example mapping rule as provided later in the chapter and click Next.

11. Finish the configuration and load it into the Tivoli Federated Identity Manager run time.

Before adding the SAP system as a federation partner, you have to export the information metadata and some configuration data from the SAP system.

**SAP system: Activating security session manager**

Use your SAP GUI client to connect to the SAP system:

1. Log in as administrator.
2. Open transaction SICF_SESSIONS. Here you activate the client.
3. Switch to transaction SAML2 and configure SAML2.0 support. First click Enable SAML 2.0 Support.
   
   This action opens a browser window and starts a guided procedure.
4. Enter a provider name and click Next. Choose any provider name that you want or that matches your company definitions.
5. Accept the default Clock Skew Tolerance and click Next.
   
   The SAP server as the provider is able to serve different consumer bindings. Even we will only use HTTP Artifact. We can accept the default to support all binding types on that page.
6. For the initial configuration we choose the default Manual Identity Provider Discovery selection for debugging and testing purposes. This reflects in an additional login screen when accessing the server.
7. Finish the configuration.

The SAML2 provider configuration is done. Now we need to export the metadata.

**Exporting SAP partner metadata**

To export the metadata

1. Stay on the Local Provider tab and click Metadata.
2. A wizard opens to select the entity descriptor and Download Metadata.
3. Store the file in your local file system and close the wizard.
**Tivoli Federated Identity Manager: Creating a partner**

Each system that will participate in the interaction needs to be added as Partner. This could be done either manually or by uploading the metadata. Using the upload process is usually faster and easier:

1. Open the Browser Window where the Federation was created and click **Add Partner**.
2. Select the downloaded SAP metadata in the local file system.
3. In the next screen select that the *Typical set of incoming SAML messages and assertions* are signed by the partner (SAP).
4. Enter again the default keystore password `testonly` and accept the default certificate name to create the certificate and validate the signature of the partner. Click **Next**.
5. For the encryption, do the same keystore selection. Click **Next**.
6. For the SSL Server Authentication for Artifact Resolution a certificate also needs to be selected. Enter `testonly` again as the password and select the testkey. Click **Next**.
7. Choose the basic authentication option and provide a valid SAP user. This lowers the complexity of accepting signed certificates. Click **Next**.
8. Accept the defaults on the next screen and choose **AES-128** as the encryption method. Click **Next**.
9. Select **Use XSL or JavaScript transformation for identity mapping**. Next select the mapping rule file and click **Finish** to finalize the wizard.
10. Enable the partner and load the new configuration changes into the Tivoli Federated Identity Manager run time.

**Exporting Tivoli Federated Identity Manager partner metadata**

Now the metadata from Tivoli Federated Identity Manager is needed in the SAP system. To download it switch to the federation page. But before export we tailor some settings for our use case:

1. Select the federation and click **Properties**.
2. We can select HTTP Artifact as binding only, because we will just use this one, but we will select this binding later in the SAP configuration.
3. In this step we only need to change the signing method for outgoing SAML assertions. Because the SAP system requires all messages to be signed, we select this as the default and save the federation by clicking **OK**. Then we load the configuration changes into the Tivoli Federated Identity Manager run time.

   Now we can download the metadata file to our local system.

4. Select the **Federation**, click **Export**, and save the file.

**SAP system: Partner configuration**

In the SAP SAML2 configuration we upload the metadata to create a new Trusted Provider:

https://saptest.test.com:44323/sap/bc/webdynpro/sap/saml2

1. Select **Add → Upload Metadata File**.
2. Select the file in your local system and click **Next**.
3. You can enter an alias name to improve the selection. Click **Next**.
4. You see that the Signature and Encryption options are already filled with default values. Accept them and click **Next**.
5. For the Single Sign-On Endpoints, select **HTTP Artifact binding**. This is always suggested in combination with an SAP NetWeaver Gateway.
6. For Single Logout Endpoints, select **HTTP Artifact Binding**. Click **Next**.
7. For the Artifact Endpoint the value is already set as the default. Accept it and click **Next**.
8. The default authentication requirements work out also, so we accept them and choose to **Finish** the wizard.
9. Before we can enable the trusted provider we need to add the NameID format. Click **Edit** and **Add** a NameID format.
10. We choose **E-mail** for the NameID format and click **OK** to close the wizard. Then click **Save**.
11. Now enable the configuration for the trusted provider.
Tivoli Federated Identity Manager: Editing the partner configuration

As the standard SAP partner configuration is working with https, we reconfigure this for this example to http.

1. Select the partner in Tivoli Federated Identity Manager and click Properties.
2. Change protocol and port to http. Keep the Assertion Consumer Service URL as https because this needs to be https.

SSL configuration/certificate exchange

To establish an SSL connection between two federation partners, both systems need to use a matching SSL certificate to validate the messages.

Tivoli Federated Identity Manager: Extracting the SSL certificates

First we extract the Tivoli Federated Identity Manager certificate:

1. Open the Integrated Solutions Console.
2. Navigate to Security, SSL certificate and key management. Here you use the Key stores and certificates link.
3. Open the NodeDefaultKeyStore by clicking the link.
4. Click the Personal certificates link.
5. Make sure that long domain names are used, as they work out better in general than short names. If long domain names are not used, create a new self-signed certificate and replace the old default one. Select the certificate and click Extract.
6. Provide a unique name and export the certificate in Binary DER data.
7. Download the certificate from the Tivoli Federated Identity Manager server file system, for example, at this UNIX path:
   `/opt/Tivoli/WebSphere/AppServer/profiles/AppSrv01/etc/TFIM_SSL_DER`

SAP system: Importing the SSL certificate

Take these steps:

1. Use the SAP GUI and open the transaction STRUSTSS02 on the SAP server.
2. Click the small Import certificate.
3. Select the certificate on your local machine and continue.
4. The certificate now is loaded into the middle area and we can choose where to add it.
5. Select the SSL node **SSL Client (Standard)** in the tree structure on the left and click **Add to certificate list**.

6. Save the new configuration.

**SAP system: Activating SSL for communication**

Take these steps:
1. Switch in the SAP GUI to transaction **SM59** and select the new HTTP Connection.
2. Switch to edit mode and set the security option for SSL to **active**.
3. Save the configuration

**SAP system: Exporting SAP certificate**

Take these steps:
1. Open the transaction **STRUSTSS02** in the SAP GUI and select the certificate by double-clicking the certificate name. The values display in the center.
2. Click **Export**.
3. Export the certificate in binary mode and save it on the local file system.
4. Copy the certificate to the Tivoli Federated Identity Manager server into a directory to which you have access.

**Tivoli Federated Identity Manager: Importing SSL certificate as signer certificate**

Take these steps:
1. In the NodeDefaultKeyStore click the **Signer certificates** link.
2. Enter the alias and local path to the file on the Tivoli Federated Identity Manager server.
3. Save the changes and validate the correct import.

**SAP system: Creating user mapping**

Open the transaction **SU01** and add the email address of an SAP user, so that the given email from Tivoli Federated Identity Manager can be mapped to an existing SAP user.
SAP system: Enabling SAML authentication for the SAP service

To be able to do quick testing, enable SAML authentication for the SAP web GUI:

1. Enter transaction SICF.
2. Open path → default_host → sap → bc → gui → sap → its → webgui.
3. Open details for the service and enable SAML.

Mapping rule file

Example 12-9 is an example of the mapping rule file.

Example 12-9  Mapping rule file

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser" version="1.0">
    <xsl:strip-space elements="*" />
    <xsl:output method="xml" version="1.0" encoding="utf-8" indent="yes" />
    <!--
    Initially we start with a copy of the document.
    -->
    <xsl:template match="@* | node()">
        <xsl:copy>
            <xsl:apply-templates select="@* | node()" />  
        </xsl:copy>
    </xsl:template>
    <!--
    This template replaces the entire Principal element with one that contains just the static email
    -->
    <xsl:template match="/stsuuser:Principal">
        <stsuuser:Principal>
            <stsuuser:Attribute name="name" type="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
                <stsuuser:Value>user@sap.test.ibm.com</stsuuser:Value>
            </stsuuser:Attribute>
        </stsuuser:Principal>
    </xsl:template>
    <!--
    This template builds a new AttributeList. This involves:
    Adding an AuthenticationMethod attribute to meet SAML requirements. We assume this is always the "password" mechanism.
    -->
    <xsl:template match="/stsuuser:AttributeList">
        <stsuuser:AttributeList>
            <stsuuser:Attribute>
                <stsuuser:NameFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</stsuuser:NameFormat>
                <stsuuser:Values>
                    <stsuuser:AttributeValue user@sap.test.ibm.com</stsuuser:AttributeValue>
                </stsuuser:Values>
            </stsuuser:Attribute>
        </stsuuser:AttributeList>
    </xsl:template>
</xsl:stylesheet>
```
12.5 Service-based single sign-on to SAP backend systems using Federated Identity Manager and SAML

In this integration scenario the goal is SAML-based single sign-on using Web Services Security with an SAP NetWeaver Application Server as the Web Services Provider (Figure 12-18).

As per Figure 12-18, this is the process flow:

1. The WS-Consumer sends a SOAP Request to WS-Provider.
2. IBM WebSphere DataPower® secures the request on behalf of the WS-Consumer based on the endpoint policy defined by the WS-Provider. DataPower requests the SAML 1.1 Token with a symmetric protection key
(Confirmation Method Holder-of-Key) from Tivoli Federated Identity Manager Security Token Service (STS) using the WS-Trust protocol.

3. DataPower receives the STS-issued token.
4. DataPower sends a secured request to the WS Provider on the SAP NetWeaver Application Server ABAP.
5. SAP NetWeaver Application Server ABAP successfully authenticates the request and sends back a response.

12.5.1 The role of security metadata in the single sign-on scenario

The security settings must be expressed and published by the web service provider in a standardized manner so that any consumer in a heterogeneous landscape can interact with the provider based on the same understanding in an automated fashion.

12.5.2 SAP Web Service configuration for single sign-on

The web service in the SAP NetWeaver Application Server ABAP, which is called by the Web Service Consumer, must be configured to accept the SAML Token sent for authentication. Based on this configuration, the new Web Service endpoint will expose these settings in its Web Services Description Language (WSDL) file. This includes the following tasks:

1. In transaction SOAMANAGER create a new service endpoint configuration with authentication settings for an STS-issued SAML Token.
2. As a result, the WSDL file exposed by the new endpoint includes the necessary metadata so that a Web Service Consumer knows how to authenticate, including:
   - Address of the STS
   - Token type to request from STS

**Note:** For detailed configuration information see *Configuring SSO/STS Scenario SAML Holder-of-key in the WS Provider AS ABAP* at the SAP Help Portal at the following location:

12.6 Integrate SAP into SOA by federating the SAP login ticket

This section describes a solution that enables identity propagation from SAP Web Service clients to products from other vendors. It allows organizations that are heavily invested in SAP solutions to reuse their infrastructure in SOA projects.

The solution uses the IBM WebSphere DataPower XML Firewall in conjunction with the IBM Tivoli Federated Identity Manager Security Token Service to map the proprietary SAP identity token to an open standards token such as SAML. This augments the SAP Web Service client functionality and allows for securing web services requests sent to third-party products.

12.6.1 Introduction

Service-oriented architecture (SOA) has now been accepted at the executive level in many organizations. Nearly all SOA roadmaps attempt to lower operational costs by reusing existing infrastructure. As a result, many vendors are updating existing product suites with SOA integration points, but sometimes these updates do not allow for full integration with other systems. This is especially true when it comes to identity propagation and auditing, which are vital to any enterprise. Without identity propagation and auditing there is no accountability, and this can lead to undiscovered fraud or not being able to produce evidence of fraud. Using infrastructure to manage identity is essential. Without this infrastructure, security logic is required in the services. Externalizing the security logic improves the return on investment for SOA by increasing flexibility and reducing development effort.

Organizations that rely heavily on SAP infrastructure must meet and overcome these challenges. This integration explains a technique for extending the current SAP Web Services capabilities. Because the SAP architecture relies heavily on the SAP identity token for identity propagation, trying to expand functionality beyond the SAP identity boundary is challenging. This section explains how to overcome this challenge by using IBM products in conjunction with the WS-Security and WS-Trust standards. Using industry standards allows for integration with many vendors offering SOA products including IBM, Microsoft, and Oracle. Another advantage of this technique is that it allows for identity and token mapping to be handled by middleware infrastructure, leaving application developers to focus on their core responsibilities.

The technique used in this integration uses the WebSphere DataPower XML Firewall in conjunction with Tivoli Federated Identity Manager. The DataPower appliance is placed inline and acts as a proxy. During processing it performs a
WS-Trust call to the Tivoli Federated Identity Manager Security Token Service (STS) to exchange the SAP identity token, which was sent as a cookie, for a new token. This new token is placed in the Web Service request as a WS-Security header and forwarded to the service.

Figure 12-19 illustrates a high-level view of the solution.

This solution architecture includes these advantages:

- It integrates SAP systems into a SOA environment by converting requests into open standards-compatible messages.
- It provides extensible design that reacts to future requirements.
- It requires minimal changes to existing infrastructure.
- It propagates identity from end-to-end, allowing for authorization and auditing at every node.

### 12.6.2 SAP identity representation

The SAP identity is otherwise known as the *SAP login ticket*. It is the authentication cornerstone of every SAP application. The identity is stored in an HTTP cookie named MYSAPSSO2. By default, the cookie is created as a domain cookie, which limits its availability to servers that are located in the same DNS domain as the issuing system.

The format of the identity is proprietary to SAP systems. However, an API is available to custom non-SAP applications that allow for the consumption of the identity. The identity can only be issued by an SAP server.
Validation of the identity between SAP systems is based upon PKI techniques. That is, the identity contained in the cookie is signed using the private key of the issuing system. The public key of the issuing system is shared with the system that consumes the identity.

Figure 12-20 illustrates a typical use of the SAP identity in an SAP environment. In this use case, a client application (for example, a browser) makes use of multiple SAP servers to perform its functionality. All communication is performed over HTTP, allowing the identity to be passed as a cookie in the HTTP headers.

This is the process flow illustrated in Figure 12-20:

1. The SAP client application requests a resource on an SAP server, providing appropriate authentication credentials.
2. The SAP server is configured as an SAP identity issuer. After authentication, the SAP identity is issued in the response.
3. The SAP client application requests a resource from another SAP server. The SAP identity is sent with the request and is consumed by the SAP server, configured as an identity consumer.
4. The second SAP server might require the use of a web service. It sends the SAP identity to the web service, which is configured for consumption of the SAP identity.
Identity propagation challenges
An SOA solution solely based on the SAP identity poses a number of challenges. These challenges are outlined here:

- Not based on open standards
  - Without using open standards, the identity consumer must understand the format of the identity.
  - The identity consumer requires a separate interface for the consumption of SAP identities, rather than a single open standards-based interface.
  - The supported options for open standard identity representation are currently limited to UsernameToken and X.509 Certificates (in a BinarySecurityToken) in SAP applications.
  - To make use of open standards-based identity, for example, a UsernameToken (an identity containing a username and password), the SAP applications require redesign of their web service proxy implementation.

- Potential loss of accountability
  Currently, the open standards-based options in SAP require the configuration of a “service account,” that is, a single account is used for all client requests. This results in the loss of the caller’s identity.

- Limited to HTTP
  When using the SAP identity, Web Services must use HTTP for message transmission, because the SAP identity can only be passed as an HTTP cookie.

- Limited to a single DNS domain
  - The SAP identity can only be shared amongst servers in the same DNS domain because the identity is contained in an HTTP cookie.
  - The ability to federate beyond the organizational boundary is limited.
Chapter 12. Access management use cases

12.6.3 SOA identity solution

IBM provides a number of technologies that enable you to extend your current investment in the SAP identity. The technologies promote these attributes:

- **Open standards**
  
  Supports many open standards, including SAML, Liberty, and WS-Security extensions.

- **Extensible platform**
  
  Allows for the issuing, exchange, and consumption of custom identities written as Java modules.

- **End-to-end identity propagation**
  
  Using message-level identity propagation via open standards allows the identity to reach the end service.

The products that provide these technologies are described below.
The Tivoli Federated Identity Manager Security Token Service

The Tivoli Federated Identity Manager Security Token Service (STS) is a WS-Trust compliant web service. The STS allows for security tokens to be validated, exchanged, or issued. It provides an extensible mechanism for creating, consuming, or exchanging WS-Security tokens.

More information: The WS-Trust specification is maintained by the OASIS group and is contributed to by many vendors. For further reading see the WS-Trust specification on the OASIS site at the following location:


The Tivoli Federated Identity Manager STS is built around chains that act as assembly lines that handle requests. The chain that handles the request is determined from the request's details, such as the URL and the AppliesTo and Issuer attributes. These chains are made up of modules that validate, map, or issue tokens. There is a standard workflow for these modules, which is to validate → map → issue. This method allows for tokens to be exchanged for new tokens while being able to customize attributes and their values. An authorization step is sometimes placed between the validation and map steps.
Figure 12-22 provides a basic illustration of the workflow in the Tivoli Federated Identity Manager STS.

![Figure 12-22 Basic illustration of the Tivoli Federated Identity Manager STS workflow](image)

**WebSphere DataPower SOA Appliances**

The WebSphere DataPower SOA Appliances are a family of devices that are built specifically for handing XML, SOAP, and other SOA technologies. They are designed to act as a proxy between web service clients and servers. They provide wire-speed XML validation and transformation. The XS40 and XI50 models offer authentication, authorization, and audit capabilities and are traditionally deployed as part of an ESB or in the DMZ.

**More information:** There are many more features of these devices that are beyond the scope of this publication. For more information about WebSphere DataPower SOA Appliances see the following website:

12.6.4 Solution architecture

Using the IBM technologies described above, the solution allows for the exchange of the SAP identity token into an industry standard token. In the scenario illustrated in Figure 12-23, we exchange the SAP identity token for a SAML assertion. The solution involves placing a DataPower device between the web service client, an SAP application, and the web service, such as one hosted on WebSphere Application Server.

Figure 12-23 illustrates the solution architecture making use of IBM technologies.
Process flow
Using Figure 12-24 as a reference, this is the process flow:

1. Using a client application, the user authenticates to the SAP server. This can be done using any method supported by the SAP server, for example, username+password, SPNEGO, and so on.

2. While processing a client application request, the SAP server is required to make use of a web service. The web service client, which is located on the SAP server, sends a web server request to the DataPower device (see Figure 12-24, Label 1). This request is to a web service, but does not contain a WS-Security header. Instead, the request contains the SAP identity token as an HTTP cookie.

![Figure 12-24 Process flow through DataPower](image)

3. DataPower processes the request by pulling the SAP identity token out of the cookie and packages it into a BinarySecurityToken (BST) (Figure 12-24, Label 2). DataPower then performs a WS-Trust call-out to the Tivoli Federated Identity Manager STS (Figure 12-24, Label 3). The call-out requests a security token in exchange for the SAP identity token. The request from DataPower to Tivoli Federated Identity Manager is a WS-Trust RequestSecurityToken (RST) request. The SAP identity token is represented as a BinarySecurityToken (BST) in the wst:Base element of the RST. Using a BST element to hold the SAP identity token allows for the Base64-encoded token to be sent without being edited.
4. The Tivoli Federated Identity Manager STS decodes the SAP identity token, optionally maps the identity, creates a new token (that is, a SAML assertion), then responds to DataPower with the new token (Figure 12-24 on page 351, Label 4). The response sends the new token as part of the RequestSecurityTokenResponse (RSTR).

5. DataPower then substitutes the new token in a WS-Security header (Figure 12-24 on page 351, Label 4) and forwards the request with the new token to the web service (Figure 12-24 on page 351, Label 5).

Refer to the WS-Trust specification for specific details on BSTs, RSTs, and RSTRs at the following location:


12.6.5 Configuration

This section provides an overview of how to configure IBM technologies to implement the solution. Note that specific detail has been provided where possible. However, this section does not provide step-by-step instructions, as doing so is outside the scope of this publication.

SAP Web service client

The SAP Web service client must be configured to send the SAP identity token as a cookie. This is done using the SAP NetWeaver Developer Studio. Specifically, configure a logical port of the Web service client proxy to use HTTP authentication and select Use SAP login ticket. Additionally, the Web service client proxy logical port should be configured to communicate with DataPower as the end point.
Figure 12-25 illustrates the selection of the SAP login ticket for a web service developed using the SAP NetWeaver Developer Studio.

![Figure 12-25](image)

**Note:** Refer to SAP documentation for complete details about how to build, configure, and deploy an SAP Web service client application on an SAP NetWeaver Application Server - Java Edition:


**DataPower**
At the following IBM DeveloperWorks URL you can obtain a DataPower configuration example for this solution:

In this implementation, an XML Firewall service is used to process the request. The XML Firewall contains a AAA policy. The policy is what performs the call-out to the Tivoli Federated Identity Manager STS and places the WS-Security token into the Web service header. Custom XSL transforms are used to perform these actions.

Figure 12-26 illustrates the XML firewall containing a AAA policy.

Figure 12-26  DataPower XML firewall AAA policy
Figure 12-27 to Figure 12-30 on page 358 highlight pertinent configuration options of the AAA policy configuration.

![Diagram of DataPower X150 configuration interface](image)

**Figure 12-27  Selecting the identification method and cookie name**
Figure 12-28  Specifying the custom XSL template
Figure 12-29  Defining how to extract the resources
Figure 12-30  Specifying post processing parameters

The sample transforms are example code only and should be refined before being placed into production. The XSL transforms have parameters for the Tivoli Federated Identity Manager STS end point and the Issuer and AppliesTo parts. These must be set correctly with the correct environment variables. If they are not correct, Tivoli Federated Identity Manager cannot be contacted or will not know how to handle the request.

The dp_aaa_wstrust_exchange_sap.xsl file (Figure 12-28 on page 356) is the XSLT used to call out to Tivoli Federated Identity Manager. It is used as a custom
authentication step in a AAA action. The token (if successfully attained) can be retrieved using the following Xpath statement (while still in the AAA action):

```
/container/*[local-name()='credentials']/*[local-name()='entry'] [@type='custom']/*
```

Later in the AAA action the `dp_aaa_wstrust_postprocessing.xsl` file (Figure 12-30 on page 358) is the XSLT used to insert the new token into the ongoing request. It is used as a custom post processing step.

**Tivoli Federated Identity Manager STS**

The Tivoli Federated Identity Manager STS must be configured with a custom trust chain. This chain has the Tivoli Federated Identity Manager SAP token module instance in *validate* mode, an XSL transform mapping module instance in *map* mode, and a SAML 2.0 token module instance in *issue* mode.

Figure 12-31 illustrates the configuration of a sample Tivoli Federated Identity Manager STS module chain.

*Note:* Figure 12-31 illustrates a chain using a UsernameTokenSTSModule, instead of a SAMLTokenSTSModule.
To extract and validate the SAP identity in the incoming request from DataPower, IBM has developed and supports a Tivoli Federated Identity Manager STS Module. See 11.2, “Security Token Service trust module for SAP login ticket” on page 272, for more information.

If you are using the sample files provided, the chain should be configured with the Issuer and AppliesTo set to http://issuer/sap and http://appliesto/sap, respectively. Tivoli Federated Identity Manager uses these two values to determine which trust chain should be invoked to handle the request. Therefore, these values must match with those provided to the WS-Trust XSL Transform on DataPower.
Figure 12-32 illustrates the properties of a sample trust service chain.

**Note:** The values used by the sample trust chain are different from those used in the supplied sample files.

12.6.6 Summary

The reuse of existing systems in SOA projects is good practice but creates integration issues. One of these issues is that many product suites make it hard to propagate identity to other systems. Because identity management in a SOA environment should be part of the infrastructure, this means that integration points need to be made available by the infrastructure. This allows application
developers to focus on business logic, still have accountability, and audit in the environment.

This integration explains how an organization that has heavily invested in SAP infrastructure can propagate identity using IBM infrastructure. The solution presented in this chapter uses two IBM products in conjunction:

- The WebSphere DataPower SOA Appliances
- The Tivoli Federated Identity Manager

These products provide an extensible design that can react to future requirements while only requiring minimal changes to existing infrastructure.

12.7 Tivoli Access Manager for e-business sample use case scenarios and best practices

This chapter provides additional information about the integration between IBM Tivoli Access Manager for e-business and SAP NetWeaver Application Servers ABAP and JAVA.

The section is broken into two subsections:

- Sample use cases
  The sample use cases provide information about how to integrate IBM Tivoli Access Manager for e-business and SAP NetWeaver Application Server within an organization at an architectural level. It outlines which integrations should be deployed and provides additional insight into how these integrations will affect the company.

- Best practices
  The best practices section provides information about how to best deploy the IBM Tivoli Access Manager for e-business and SAP NetWeaver Application Server integrations into an organization’s environment. The section discusses items such as deployment, server placement, recommended firewall configuration, and details about how the integrations can best fix an enterprise environment.

We recommend that you also read Chapter 10, “IBM Tivoli Access Manager for e-business” on page 221, before reading on. Chapter 10, “IBM Tivoli Access Manager for e-business” on page 221, provides detailed information about the Tivoli Access Manager for e-business and SAP NetWeaver Application Server integration.
12.7.1 Use cases

This section provides an architectural overview of which integrations should be deployed.

These are the use cases presented:

- “Standalone SAP NetWeaver AS ABAP environment” on page 363
- “Standalone SAP NetWeaver AS Java environment” on page 364
- “Multiple SAP NetWeaver server environments” on page 364
- “SAP NetWeaver Central User Administration ABAP environment” on page 366
- “Integration with SAP NetWeaver applications” on page 367

**Standalone SAP NetWeaver AS ABAP environment**

The Tivoli Access Manager for e-business integration for SAP NetWeaver Application Server ABAP supports a standalone landscape. The standalone landscape is a single SAP NetWeaver Application Server with no central user administration.

Figure 12-33 illustrates the components involved with the integration between Tivoli Access Manager for e-business and an SAP NetWeaver ABAP standalone.

![Figure 12-33 Access Manager for e-business integration with SAP NetWeaver AS ABAP standalone environment](image)

Section 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222, provides detailed steps to configure single sign-on to a standalone SAP NetWeaver ABAP server.
Standalone SAP NetWeaver AS Java environment

The Tivoli Access Manager for e-business integration for SAP NetWeaver AS Java supports a standalone SAP NetWeaver landscape.

Figure 12-34 illustrates the components involved with the integration between Tivoli Access Manager for e-business and an SAP NetWeaver AS Java standalone.

![Diagram](image)

Figure 12-34 Access Manager for e-business integration with SAP NetWeaver AS Java standalone

Section 10.2, “Integration with SAP NetWeaver AS Java” on page 228, provides detailed steps to configure single sign-on to a standalone SAP NetWeaver AS Java.

Multiple SAP NetWeaver server environments

As identified in the above two sections, SAP NetWeaver comes in two forms:

- SAP NetWeaver Application Server ABAP
- SAP NetWeaver Application Server Java

Many deployments of SAP NetWeaver Application Server within an organization’s environment contain either multiples of the same type of server or a combination of both.

Tivoli Access Manager for e-business WebSEAL allows the creation of multiple junctions with multiple back-end servers. This means that a single Tivoli Access Manager for e-business WebSEAL environment can manage the authentication and authorization for multiple SAP NetWeaver servers independent of the type.
Figure 12-35 illustrates an environment that contains multiple SAP NetWeaver servers and how they can be placed behind Tivoli Access Manager for e-business WebSEAL.

Important: It is important to note that the steps outlined in 10.1, “Integration with SAP NetWeaver AS ABAP” on page 222, and 10.2, “Integration with SAP NetWeaver AS Java” on page 228, are specific to the configuration of the SAP NetWeaver Application Servers ABAP and that Java and must be completed for each server.

Alternatively, 10.5, “Single sign-on for SAP NetWeaver AS ABAP with WebSEAL in conjunction with SAP NetWeaver AS Java” on page 243, provides another way of integrating Tivoli Access Manager for e-business and SAP NetWeaver servers, optimizing the best integration points from each SAP NetWeaver platform. This integration also removes the need to manage global single sign-on credentials.
for the SAP NetWeaver Application Server ABAP, which reduces the administration effort.

Another alternative is to deploy multiple Tivoli Access Manager for e-business WebSEAL servers or instances to manage the different SAP NetWeaver servers. Figure 12-36 illustrates this environment's configuration.

![Figure 12-36 Multiple Tivoli Access Manager for e-business integration deployed within a multiple SAP NetWeaver server environment](image)

**SAP NetWeaver Central User Administration ABAP environment**

SAP NetWeaver Central User Administration allows the management of user accounts across the SAP NetWeaver landscape from a single management console. In regard to access management, an SAP Central User Administration environment has minimal impact. The main consideration in regard to deploying the Tivoli Access Manager for e-business integrations into this environment is the synchronization between the user registries.
Integration with SAP NetWeaver applications
The integrations provided by IBM integrate directly with the core of the SAP NetWeaver application servers. At times some SAP NetWeaver applications might need additional configuration to allow single sign-on to occur. In these cases the steps to complete the integration are left to the deploying administration team. Also see 12.7.2, “Best practices” on page 367, for specific implementations for integrations of Tivoli Access Manager for e-business with SAP CRM and SAP SRM applications.

12.7.2 Best practices
In this section we present known best practices and recommendations related to the deployment of IBM and SAP access management solutions. The discussion includes the following topics:

- “Deployment of Tivoli Access Manager for e-business” on page 368
- “User name synchronization” on page 368
- “Server placement” on page 369
- “Firewall considerations” on page 370
- “Security considerations” on page 371
- “High availability” on page 372
- “SAP hook authentication” on page 373
- “Performance issues” on page 373
- “SAP version mixture” on page 374
- “SAP CRM/SRM punch-out catalogues” on page 375
- “Inconsistency of SAP definition of pages and MIME objects” on page 377
- “Correct setting of HTTPURLLOC” on page 377
- “Load balancing and session management” on page 377
Deployment of Tivoli Access Manager for e-business

Table 12-2 presents best practices and recommendations for the deployment of Tivoli Access Manager for e-business.

Table 12-2  Best practices and recommendations

<table>
<thead>
<tr>
<th>Consideration: Which operating system should Tivoli Access Manager for e-business be deployed?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> Tivoli Access Manager for e-business supports a wide range of operating systems. We recommend that it is deployed on an AIX system to fully take advantage of the optimizing capabilities that it offers. In addition, IBM will be able to assist more with performance tuning and security configuration.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consideration: We have an increasing reliance on virtual hosted machines. Can I deploy Tivoli Access Manager for e-business into the virtualized environment?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> The deployment of Tivoli Access Manager for e-business can be supported in a virtualized environment, but we do not recommend that. Degraded performance has been reported in virtualized environments. It is critical to complete testing between a physical environment and a visualized one to ensure that the selected one is capable of meeting the company’s business needs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consideration: What are the hardware considerations for hosting Tivoli Access Manager for e-business?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> When considering hardware to deploy Tivoli Access Manager for e-business it is important to first understand the amount of users and their usage. After you have this information it will be important to ensure that the levels of CPU, random access memory, and network connectivity are considered. Another consideration that needs to be taken into account is the company’s future growth potential. This way hardware can be purchased to allow for this growth. For additional information about hardware recommendations contact the IBM account team.</td>
</tr>
</tbody>
</table>

User name synchronization

Table 12-3 presents best practices and recommendations for user name synchronization.

Table 12-3  Best practices and recommendations

<table>
<thead>
<tr>
<th>Consideration: What recommendations does IBM have for ensuring the synchronization between account names between Tivoli Access Manager for e-business and SAP NetWeaver Application Server?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice:</strong> It is recommended that IBM Tivoli Identity Manager is used across the entire enterprise. This central management of user identities ensures that the required users are created on all required systems. An alternative is to use IBM Tivoli Directory Integrator to complete the synchronization between the two servers.</td>
</tr>
</tbody>
</table>
Server placement

Table 12-4 presents best practices and recommendations for the server placement.

Table 12-4  Best practices and recommendations

<table>
<thead>
<tr>
<th>Consideration: When deploying Tivoli Access Manager into my company's enterprise where should all the components be placed and why?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Best practice: Each Tivoli Access Manager for e-business environment is unique depending on the different business requirements that each company has. Figure 12-37 on page 370 illustrates a basic architectural overview of a simple environment with Tivoli Access Manager for e-business, including the additional SAP NetWeaver Application Servers. It is important to note the following server placements:</th>
</tr>
</thead>
</table>

1. There are two separate Tivoli Access Manager for e-business WebSEAL servers. One is placed within the company's DMZ and the other is placed in the intranet. The reason for this is that you do not want the external users to need access to the intranet, as this would give them more access than they should have to other systems. The Tivoli Access Manager for e-business WebSEAL placed within the intranet is used by internal company employees. This ensures that all authentication and authorization requests are managed in a central place. It also means that the internal employees do not need direct access to Secure Zone 1.

2. The installation of the Tivoli Access Manager for e-business Proxy Policy Server within the intranet ensures that no direct connects are made directly against the policy server itself. All requests from both WebSEALs will be made via the proxy server, preventing the machines from needing direct access.

3. The Tivoli Access Manager for e-business Proxy Policy Server has been placed into Secure Zone 1 to ensure that it is secured from all direct connects coming from in from the internet or the intranet. In this case only connects from the Tivoli Access Manager for e-business Proxy Policy Server are allowed to pass to the policy server.

4. Both the SAP NetWeaver Application Server ABAP and Java have also been placed into Secure Zone 1. This is for the same reason as the placement of the Tivoli Access Manager for e-business Policy Server. Placing these servers within this zone prevents all direct connections, and only requests via Tivoli Access Manager for e-business WebSEAL are allowed to be passed onto the SAP NetWeaver servers.

5. Finally, all the user repositories have been placed into Secure Zone 2. A company's user data is one of the most critical pieces of information. We have been placed deeper into the secured zones to ensure that no access is granted from either the internet or intranet zones. Only requests coming from Secure Zone 1 products will be allowed to pass into Secure Zone 2. All other requests will be denied.
Firewall considerations

Table 12-5 presents best practices and recommendations for firewall considerations.

Table 12-5  Best practices and recommendations

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>What considerations need to be made when placing firewalls into a Tivoli Access Manager for e-business and SAP NetWeaver Application Server environment?</td>
<td>In Figure 12-37 on page 370 there is a line between each network zone that separates the network zones and represents the location where a firewall would be placed within environment. Each firewall needs to be configured to grant only requests from systems that are authorized to send them. In addition, the firewalls need to be configured to allow only requests on specific ports. A final note is that all outbound requests from a more secured zone need to be denied. This prevents any data leakage from a more secure zone into a less secure one. For additional information about specific firewall setting consult the Tivoli Access Manager for e-business infocentera or contact your IBM account team.</td>
</tr>
</tbody>
</table>

Security considerations
Table 12-6 presents best practices and recommendations for security considerations.

Table 12-6  Best practices and recommendations

<table>
<thead>
<tr>
<th>Consideration: Are there any specific considerations for security within a Tivoli Access Manager for e-business and SAP NetWeaver Application Server environment?</th>
</tr>
</thead>
</table>
| **Best practice**: The most important security consideration for security within this environment is to ensure that secure communications occur between each of the servers. The following considerations need to be reviewed:  
  - Tivoli Access Manager for e-business WebSEAL should be configured to access HTTPS connections from external and internal users.  
  - Communication between the Tivoli Access Manager for e-business and SAP NetWeaver back-end servers also needs to be confirmed to use HTTPS connections only. This is most important for the integration for the SAP NetWeaver application ABAP server, as the user global single sign-on username and password are inserted into the request as basic authentication headers and are viewable over the network as clear text.  
  - It is important to secure the information between the servers within Secure Zone / and their user repositories. Though these systems will be extremely hard to access unless getting access within the specific zones, there is still a chance that these zones could be compromised. Securing the network traffic between these servers ensures that all information across the architecture is encrypted, removing any potential chance of information loss via network sniffing. |

<table>
<thead>
<tr>
<th>Consideration: Is there anything else that needs to be considered when securing a Tivoli Access Manager for e-business and SAP NetWeaver Application Server environment?</th>
</tr>
</thead>
</table>
| **Best practice**: The security of the operating system running on each server must be considered when considering security. If the server within the environment is easily compromised, then network security will have minimal effect.  
  The deployment on an AIX system is recommended to fully take advantage of the optimizing capabilities that it offers. If AIX is used, IBM will be able to assist with recommendations about securing this environment. For all other operating systems the specific owning vendors need to be engaged.  
  Additionally, securing the SAP NetWeaver Application Servers must be considered. Again, in this case we recommend that SAP support is engaged to provide assistance. |
**High availability**

Table 12-7 presents best practices and recommendations for high availability.

**Table 12-7  Best practices and recommendations**

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice: Tivoli Access Manager for e-business WebSEAL allows the creation of multiple replicated instances to allow for a highly available environment. Figure 12-37 on page 370 illustrates an environment with multiple instances. When using multiple instances, one of the major factors to consider is user session management across all the Tivoli Access Manager for e-business WebSEAL instances. Tivoli Access Manager for e-business WebSEAL provides session failover, where a cookie presented by one WebSEAL instance can be consumed by another. Additional information about failover cookies can be found at the following URL: <a href="http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am611_webseal_admin388.htm?path=3_3_1_6_2#chap-failover">http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am611_webseal_admin388.htm?path=3_3_1_6_2#chap-failover</a> An alternative to the above Tivoli Access Manager for e-business provides the Session Management Server. The Session Management Server manages all user sessions across multiple realms within a clustered WebSEAL environment. The usage of the Tivoli Access Management Server is the most complex, but it is extremely critical when configuring a highly available Tivoli Access Manager for e-business environment. More information about the Tivoli Access Manager for e-business Session Management Server can be found at the following URL: <a href="http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am611_sms_admin.pdf">http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am611_sms_admin.pdf</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Consideration</td>
<td>Best practice: We recommend that the SAP support team be engaged for assistance.</td>
</tr>
<tr>
<td>Consideration</td>
<td>Are there any considerations for the usage of load balances within a Tivoli Access Manager for e-business?</td>
</tr>
</tbody>
</table>
**Best practice**: The following IBM DeveloperWorks article provides initial information about configuring load balancers within a Tivoli Access Manager for e-business environment:


For additional information about Tivoli Access Manager for e-business within a highly available environment, contact your IBM account team representative.

**SAP hook authentication**

Table 12-8 presents best practices and recommendations for SAP hook authentication.

**Table 12-8  Best practices and recommendations**

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certain SAP applications call out to another SAP server for additional authentication. When the user accesses specific applications on the backend SAP NetWeaver server, the server sends a redirect to an external SAP server. What considerations need to be made when an application that makes these external callouts is encountered?</td>
<td>If an application is encountered that uses hook authentication, we highly recommend that all junctions created within the Tivoli Access Manager WebSEAL instance are virtual host junctions. One of the major problems when a backend server calls out to another server is that the request has no way of returning, as all requests go directly through Tivoli Access Manager for e-business WebSEAL. Creating virtual host junctions ensures that the request is able to return to the back-end server via Tivoli Access Manager for e-business WebSEAL.</td>
</tr>
</tbody>
</table>

**Performance issues**

Table 12-9 presents best practices and recommendations for performance issues.

**Table 12-9  Best practices and recommendations**

<table>
<thead>
<tr>
<th>Consideration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>What performance steps do I need to take with regard to Tivoli Access Manager for e-business and the integration between SAP NetWeaver Application Servers?</td>
<td></td>
</tr>
</tbody>
</table>
**Best practice**: There are many ways to improve the performance of Tivoli Access Manager for e-business. One of the main ways is to ensure that correct indexing has been configured in the LDAP server.

More information about performance tuning Tivoli Access Manager for e-business can be found at the following URL:


With regard to the integration itself, network connectivity and performance are critical. Ensure that all network lines are working at an optimal speed and that there are no bottlenecks within the environment. Special notice should be taken with regard to the routes taken between the Tivoli Access Manager for e-business and SAP NetWeaver Application Servers.

**Consideration**: I have applied performance tuning to Tivoli Access Manager for e-business and completed network testing within my environment, but the users are still reporting that their requests are taking a long time to return to the browser. What is wrong with Tivoli Access Manager for e-business?

**Best practice**: If you have completed all performance tuning with regard to the Tivoli Access Manager for e-business and your users continue to observe degraded performance, we recommend that you review the current performance of the SAP NetWeaver Application Servers. Contact SAP support to assist in this matter.

**SAP version mixture**

Table 12-10 presents best practices and recommendations for SAP version mixtures.

Table 12-10  **Best practices and recommendations**

**Consideration**: Are there any considerations with regard to the support of older and newer versions of SAP NetWeaver Application Server?
### Best practice

As long as the SAP NetWeaver Application Server version is specified within the officially released integration guides, then IBM will support the configuration.

If a newer version is encountered, the Tivoli Access Manager for e-business product management team needs to be notified of the request by submitting an IBM Market Request.

For assistance with creating an IBM Market Request, contact your IBM account team. After the Market Request has been submitted it will be analyzed, and if the product management agrees that this is a strategic requirement, it will be added to a future release of the integration.

IBM is unable to support any older version of SAP NetWeaver that SAP has stopped support.

---

### SAP CRM/SRM punch-out catalogues

Table 12-11 presents best practices and recommendations for SAP CRM/SRM punch-out catalogues.

<table>
<thead>
<tr>
<th>Consideration</th>
<th>One of the issues found with SAP CRM and SAP SRM applications and Tivoli Access Manager for e-business implementations is that of punch-out catalogues. In this scenario the return address of the SAP system (HOOK_URL) is embedded in both the web page and the URL of the request.</th>
</tr>
</thead>
</table>
Best practice: To fix this issue the following notes were used:

- Tivoli Access Manager for e-business 6.0 and 6.1
  Tivoli Access Manager for e-business WebSEAL filtering of hostnames in URLs changed in Tivoli Access Manager for e-business 6.0.
  A compatibility option is provided to support the 5.1 behavior.
- Tivoli Access Manager for e-business 5.1
  In Version 5.1 of Tivoli Access Manager for e-business, if a junction was created with the -v flag, then hostnames that matched the server specified with either the -h flag or the -v flag were filtered and the corresponding link converted to use the junction point.
  In Version 6.0 of Tivoli Access Manager for e-business, only hostnames that match the server specified with the -v flag are filtered.

The 5.1 behavior was incorrect. If WebSEAL is to filter links, it must be able to determine to which junction it should map the link. To do so, there must be a one-to-one mapping between the hostname specified in the URL and a junction point.

In the case of -v junctions, however, it is possible to create multiple junctions using the same hostname and different virtual hostnames. In this scenario, there is a one-to-many mapping between the hostname and junction points. In such a case, it is not possible for WebSEAL to accurately determine which junction point to use when mapping links for the hostname. For this reason, Version 6.0 of WebSEAL does not use the hostname, but only the virtual hostname when performing a mapping.

If, for any reason, the 5.1 behavior is desired, the only scenario in which the mapping issue will not cause a problem is if multiple junctions with the same hostname and different virtual hostnames do not exist. Otherwise, the behavior will be unpredictable, with WebSEAL mapping the hostname to different junction points arbitrarily.

To enable the 5.1 behavior, modify the WebSEAL configuration file as follows:

```
[junction]
hostname-aliasing=yes
```

With this fix pack in place you should be able to set the -h and -p flags to the internal hostname that SAP is using (for example, xxx.local) and then set the external hostname port with the -v parameter.
Inconsistency of SAP definition of pages and MIME objects

Table 12-12 presents best practices and recommendations about inconsistency of SAP definition of pages and MIME objects.

**Table 12-12  Best practices and recommendations**

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP does not always get the mime-type set correctly on all pages and web objects.</td>
<td>Make sure that all SAP Notes are installed to correct these errors. This might not be as much of an issue with later SAP releases and versions.</td>
</tr>
</tbody>
</table>

Correct setting of HTTPURRLLOC

Table 12-13 presents best practices and recommendations for the correct setting of HTTPURRLLOC.

**Table 12-13  Best practices and recommendations**

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sometimes creative settings need to be made to the HTTPURRLLOC table to ensure that the URL provided by SAP is consistent with the infrastructure, especially if there is a hardware load balancer between Tivoli Access Manager for e-business WebSEAL and the user.</td>
<td>Setting the hostname to the virtual hostname of the load balancer is the only way to get the service working. This is most prevalent in SAP areas like Learning Solution and Adobe Forms.</td>
</tr>
</tbody>
</table>

Load balancing and session management

Table 12-14 presents best practices and recommendations for load balancing and session management.

**Table 12-14  Best practices and recommendations**

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Best practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do you ensure correct load balancing and session management?</td>
<td>To ensure correct load balancing and session management, always use an SAP Webdispatcher between the Tivoli Access Manager for e-business WebSEAL and the Application Server.</td>
</tr>
</tbody>
</table>

12.8 Tivoli Access Manager for Enterprise Single Sign-on SAP use cases

This section provides an architectural overview of which integrations should be deployed for specific use cases.
The use cases presented include:

- Installation of Tivoli Access Manager for Enterprise Single Sign-on AccessProfile for SAP GUI/SAP WebGUI (SAP GUI for HTML) using the Tivoli Access Manager for Enterprise Single Sign-on AccessProfile for Microsoft Internet Explorer/SAP WebGUI using the Mozilla Firefox profile
- Installation check for Tivoli Access Manager for Enterprise Single Sign-on AccessProfiles for SAP solutions
- Deployment scenarios

12.8.1 Installation

To install the Tivoli Access Manager for Enterprise Single Sign-on AccessProfile for SAP GUI/SAP WebGUI (SAP GUI for HTML), take these steps:

1. Download the package file that contains the profile\(^7\).
2. Unpack the package file to read any included documentation.
3. Extract the correct .eas file containing the desired AccessProfile.

---

\(^7\) The standard Tivoli Access Manager for Enterprise Single Sign-On AccessProfile for SAP solutions can be obtained by downloading the latest Tivoli Access Manager for Enterprise Single Sign-On Enterprise Software Profile Bundle from the IBM support site (component ID 5724N70IF). This is the direct link to this resource:

https://www.ibm.com/support/docview.wss?uid=swg24029132
4. Using AccessStudio, upload the AccessProfile to the IMS server (Figure 12-38).

![Figure 12-38](image)

5. For the SAP GUI AccessProfile, follow the instructions included in the profile package regarding SAP environment settings.

6. For the SAP GUI AccessProfile, if random passwords are required for password change, follow the instructions included with in the profile package to enable random password generation.

### 12.8.2 Installation check

After it is installed, test the AccessProfile for correct operation:

2. Log on to the SAP NetWeaver Application Server using an existing SAP user ID.
3. The Tivoli Access Manager for Enterprise Single Sign-On AccessAgent prompts you to store the SAP user logon credentials.
4. After the credentials are stored, check the Wallet to ensure that the credentials are present.
12.8.3 Deployment scenarios

In this section we discuss deployment scenarios.

**Single SAP NetWeaver Application Server scenario**

The Tivoli Access Manager for Enterprise Single SignOn AccessProfile for SAP GUI and SAP GUI for HTML (WebGUI) browser profiles support any number of SAP NetWeaver AS, including a standalone landscape. Figure 12-39 shows the standalone landscape with a single SAP NetWeaver AS.

![Figure 12-39   Single SAP NetWeaver AS scenario](image)

**Multiple SAP NetWeaver AS scenario**

After it is installed, the AccessProfiles can be used by any AccessAgent system in the same domain after synchronization with the Tivoli Access Manager for Enterprise Single Sign-On IMS.
Multiple SAP NetWeaver AS logons can be stored in the Tivoli Access Manager for Enterprise Single Sign-On AccessAgent Wallet, enabling single sign-on to multiple SAP systems (Figure 12-40).

Figure 12-40  Multiple SAP NetWeaver AS scenario

12.9 Conclusion

This chapter provided additional information to assist the reader with their understanding of the IBM Security access management and SAP solutions and how to apply them into their enterprise. It did this by illustrating different use cases and identified which solution should be deployed. It also provided information about best practices to give the reader more insight into how the
solutions should be used. Finally, the chapter provided additional scenarios outlining extended IBM and SAP solutions.
IBM Security systems integrations: Beyond identity and access management

In this book we focused mostly on integrations between IBM Security identity and access management products and SAP application servers. In this appendix we briefly look at other IBM Security products that offer integration capabilities with SAP solutions.

We cover the following topics:

- “IBM Rational AppScan” on page 384
- “IBM InfoSphere Guardium” on page 385
- “IBM InfoSphere Guardium Encryption Expert” on page 390
- “IBM InfoSphere Optim” on page 393
- “IBM WebSphere DataPower” on page 395
- “IBM System z and mainframe security” on page 397
- “IBM Power Systems and AIX Security” on page 400
IBM Rational AppScan

When SAP applications are the backbone of your business, security vulnerabilities in those applications introduce immeasurable risk to your most critical processes and sensitive data. You trust SAP solutions for financial reporting, human resources, supply chains, customer relationship management, and more. You need a solution that reduces the risk of security breaches and data loss by identifying and remediating security vulnerabilities.

Many organizations believe securing their SAP solutions begins and ends with a segregation of duties and access rights management to limit system and application privileges appropriately. However, security vulnerabilities also present a great risk because they open the door for attacks against the applications that circumvent access controls. For example, unvalidated input is a common vulnerability for any application (including SAP applications) that allows for SQL injection attacks to access, create, change, or delete data without authorization. Imagine the consequences of a SQL injection attack against your critical SAP systems that you rely on for financial reporting or managing customer data. Security vulnerabilities are like quality defects because they occur naturally in any application development process. Organizations require tools and solutions for identifying and remediating these vulnerabilities in the development process and application management life cycle. SAP applications, both web portals and Advanced Business Application Programming (ABAP) applications, face the same type of security vulnerabilities as most other applications, such as SQL Injection. Although traditional web application security solutions can address SAP web portals, ABAP applications require SAP expertise and advanced security testing to analyze ABAP source code.

IBM Rational AppScan solutions for SAP security

The IBM Rational AppScan® suite of application security testing solutions helps automate the analysis of SAP applications—including web portals and ABAP applications—to identify security vulnerabilities and manage application risk. The Rational AppScan portfolio includes dynamic, static, and hybrid analysis application testing solutions that have proven value for the most advanced web applications. For ABAP applications, IBM has partnered with the SAP security experts at Virtual Forge GmbH to offer CodeProfiler for Rational AppScan Source Edition software, which delivers advanced static analysis of ABAP source code. IBM Rational AppScan solutions for SAP security combine these:

- Advanced security research
- SAP expertise
- Guidance on how to correct vulnerabilities
- Integration with development processes
Application Life cycle Management (ALM)
Enterprise reporting and application risk management

CodeProfiler for Rational AppScan Source Edition identifies security vulnerabilities in SAP ABAP applications and enables enterprises to eliminate SAP application risk with advanced static (white box) security testing of ABAP source code:

- Identify and remediate security vulnerabilities in your SAP applications by analyzing ABAP source code to expose security defects with static (white box) analysis.
- Empower developers to write secure ABAP applications by integrating security scanning into the ABAP Workbench and SAP user interface.
- Enforce service-level agreements for security for applications and code developed by consultants and third parties.
- Drive remediation efforts with recommended code fixes and triage results in AppScan Source Edition for a single view of all static analysis testing and results.
- Manage SAP security as part of your enterprise application risk management program by integrating with Rational AppScan Enterprise.

More information: For more information about Rational solutions for SAP see this website:


IBM InfoSphere Guardium

IBM InfoSphere® Guardium® is a database security solution for ensuring the privacy and integrity of trusted information in your data center (including solutions from SAP, PeopleSoft, IBM Cognos®, Siebel, and so on) and reducing costs by automating the entire compliance auditing process in heterogeneous environments.

Guardium can manage the entire database security and compliance life cycle. It provides a robust solution for safeguarding financial and ERP information, customer and cardholder data, and intellectual property stored in enterprise systems.

Guardium is an enterprise security platform that can prevent unauthorized or suspicious activities by privileged insiders and potential hackers. It can also monitor potential fraud by users of enterprise applications such as Oracle E-Business Suite, PeopleSoft, SAP solutions, and in-house systems. At the
same time, the solution optimizes operational efficiency with a scalable, multi-tier architecture that automates and centralizes compliance controls across your entire application and database infrastructure.

Guardium does not require changes to your databases, has only minimal impact on performance, and does not rely on native database logs or auditing utilities.

IBM InfoSphere Guardium addresses the entire database security and compliance life cycle with a unified web console, back-end data store, and workflow automation system, enabling you to take these actions:

- Locate and classify sensitive information in corporate databases.
- Assess database vulnerabilities and configuration flaws.
- Ensure that configurations are locked down after recommended changes are implemented.
- Provide 100% visibility and granularity into all database transactions, across all platforms and protocols, with a secure, tamper-proof audit trail that supports separation of duties.
- Track activities on major file-sharing platforms, such as Microsoft SharePoint.
- Monitor and enforce policies for sensitive data access, privileged user actions, change control, application user activities, and security exceptions such as failed logins.
- Automate the entire compliance auditing process (including report distribution to oversight teams, sign-offs, and escalations) with pre-configured reports for SOX, PCI DSS, and data privacy.
- Create a single, centralized audit repository for enterprise-wide compliance reporting, performance optimization, investigations, and forensics.
- Easily scale from safeguarding a single database to protecting thousands of databases in distributed data centers around the world.

Integration with SAP solutions

SAP solution landscapes can be quite complex. From an audit perspective, SAP solutions can provide system generated information to help identify basic information that you need to secure your SAP systems. But the SAP three-tier architecture makes it difficult to enforce security beyond that and to perform audits non-intrusively in SAP environments. The following examples illustrate this:

- SAP solutions use pooled database connections, so all front-end users share the same technical DB user with all permissions required to run SAP. Security and access control therefore cannot be enforced on the DB layer.
- When two distinct front-end users, Joe and Bob, share the same technical DB user, they cannot uniquely be identified at the database level for access monitoring and security policies.

With Guardium, detailed information is available about SAP solution users and their activity. This information reaches beyond SAP transaction logs and makes it easier to detect fraud and unauthorized activity. Guardium integrates with SAP solutions and SAP databases with no application changes required.

Guardium provides three types of policy rules:

- An access rule that evaluates client requests and fires on request parameters, that is, columns/tables accessed, time of day, IP address of the client, and so on.
- An extrusion rule evaluates data returned by the server, that is, to protect credit card information or VIP information.
- An exception rule evaluates exceptions returned by the server, that is, to protect against SQL injection or brute force attacks, which typically involve a lot of SQL exceptions.

When policy fires, four types of actions can be performed:

- Logging
  Log the request.
- Masking
  Mask the data in the response to protect critical information.
- Alerting
  Alert on an invalid request in real time.
- Session blocking & quarantine
  Terminate the DB connection and quarantine the user for a specific time so that you can investigate the incident.
This allows you to enforce change and access control policies for critical SAP tables and allows for drill-down of user activity to navigate easily from the incident to individual SQL statements issued by the user.

In addition, the application user identification of Guardium ensures accountability by reporting on SAP user credentials from which unauthorized operations were performed, unlike native database audit logs that only identify actions performed by the SAP generic service account.

**Guardium auditing for SAP solutions**

In addition to its security features, Guardium has predefined policies and reports for auditing SAP systems. These policies help to comply with a variety of audit requirements, for example PCI, SOX, and ISO27001. All rules can be customized easily by using a web interface. Furthermore, the InfoSphere Guardium Database Protection Knowledge Base automatically updates rules and policies according to changing SAP solution specification and audit requirements.

Guardium can be used to reduce time and effort required to demonstrate compliance with SOX, PCI-DSS, FISMA, SAS70, and data privacy regulations, by automating the entire compliance auditing process.

Guardium provides a granular, tamper-proof audit trail of all database activities performed by SAP DBAs, developers, outsourced personnel, and application users. It can mask or de-identify sensitive SAP data in test and development environments.

**Integration scenario: Monitoring transaction-based user activity**

SAP user activity is based on transactions. For example, G/L account posting is transaction FB50. The database log only shows a pooled SAP database user.

With Guardium you can map the database activity with the unique SAP user that executed the transaction.

**Integration scenario: Monitoring add users transaction (SU01)**

SU01 is an SAP transaction code that allows you to add users with SAP systems. The SAP application stores this information inside many database tables.

These are SAP database tables relating to user information:

- ADRP = Personal data like first and last names
- USR01 = User master record
- USR02 = Logon data
- USR04 = Master authorization
With Guardium you can identify and correlate user activity based on table information.

**Integration scenario: Identifying who accessed specific SAP-sensitive information**

With Guardium you can audit who accessed or changed specific records, for example, customer records. The SAP transaction code to display customer records is XD03.

Guardium can log and audit anytime someone looks at a specific customer record like *Biker GmbH*, for example, to find out that “Joe looked at ‘Biker GmbH’ customer number 10001.”

If someone deletes a customer record, it ripples through the entire SAP system. Many tables are affected by the Delete flag. SAP auditing can be useful in troubleshooting SAP transactions, because you have the full history available.

You can use Guardium to audit, for example, SAP Transaction Code XD02 (Change Customer Records) to identify who changed the data in the SAP system and also who is actually authorized.

**Integration scenario: Monitoring privileged SAP users**

SAP applications use root-like users such as SAP* and DDIC. The first action an administrator should take is to change the default passwords of these users because they are well known by the SAP community. The second step is to restrict access to these privileged users to a selected few only, or even better, to introduce privileged identity management tooling such as IBM Privileged Identity Management based on Tivoli Identity Manager.

In addition to privileged SAP users, there are database users that also allow extensive access to sensitive data. As an example, see the system-defined user “sapfcp” (SAP<SID> user). This is an operating system defined user. During authentication, DB2 (or any third-party tool that makes use of this user account) verifies with the operating system that the user/password combination entered is correct. If the user is stored in the operating system (and not externally, such as in an LDAP directory or similar), then the password of this user is managed by the operating system.

The SAP<SID> user is a critical user in every SAP ABAP environment that owns all the SAP database tables and runs most of the SQL statements. Everyone who has access to the database system and who knows the user and the password could read all data.

With Guardium you can monitor and track access to the user and his activities.
IBM InfoSphere Guardium Encryption Expert

InfoSphere Guardium Encryption Expert supports SAP software. It helps satisfy compliance requirements by encrypting all database files, along with reports and the log data. Guardium Encryption Expert encrypts databases and files in place and avoids the need to rearchitect databases, files, or storage networks. Inserted above the file system and/or logical volume layers, Guardium Encryption Expert is transparent to users, applications, databases, and storage subsystems. It requires no ABAP programming language coding and no modification to SAP software or the database, so deployments can be managed in weeks rather than months. Features and benefits include these:

- Transparent, rapid implementation
  InfoSphere Guardium Encryption Expert encrypts databases and files “in place” and avoids the need to re-architect databases, files, or storage networks. Inserted above the file system or logical volume layers, InfoSphere Guardium Encryption Expert is transparent to users, applications, databases, and storage subsystems. It requires no coding and no modification to applications or databases, and consequently deployments can be managed in weeks rather than months.

- Structured and unstructured data
  InfoSphere Guardium Encryption Expert can secure structured and unstructured data to satisfy rigorous audit requirements and provide comprehensive protection for sensitive data.

- High performance
  Benchmarking has demonstrated that the InfoSphere Guardium Encryption Expert solution has no discernible performance impact for users. InfoSphere Guardium Encryption Expert performs encryption and decryption operations at the optimal location of the file system or volume manager and consequently minimizes performance overhead. This approach leverages the I/O profile of databases by only encrypting and decrypting the storage blocks needed for a particular operation.
Centralized management

InfoSphere Guardium Encryption Expert minimizes administrative overhead with key and policy management, providing a secure, easy-to-administer method of administering encryption keys. It enables consistent and common best practices for managing the protection of both structured and unstructured data accessed.

Fine-grained auditing

InfoSphere Guardium Encryption Expert provides granular and configurable auditing and reporting of access requests to protected data, in addition to changes to policies and keys. The system's audit management reduces audit scope, integrates with existing Security Information and Event Management (SIEM) solutions, and aids compliance with industry and regulatory practices regarding the handling and protection of private and confidential information.

Scalability

Organizations can scale InfoSphere Guardium Encryption Expert in large and complex environments including thousands of systems and files.

Support for SAP applications

Enterprises must ensure that data is protected from both theft and misuse. Because SAP applications do not provide such functionality, organizations rely on the SAP partner ecosystem to provide data protection, including safeguarding data at rest or in use. Encrypting sensitive data at rest can minimize the possibility of data breaches and satisfy audit requirements.

Any strategy for securing SAP data needs to minimize the impact on SAP software and IT operations. Minimizing change to an SAP software environment allows for rapid implementation of a data security solution and avoids burdening IT with significant ongoing management costs. Burdens to consider can come in the form of changing SAP software, integration, and testing or modifying the underlying hardware topology. Considering and controlling such changes results in a higher probability of success. To the degree that changes can be avoided, a project can also roll out more quickly.

While SAP software is an exceptionally comprehensive system, the structure of SAP databases is such that sensitive data in multiple data types can be spread throughout the database. For example, the SAP ERP Human Capital Management solution might contain sensitive information about employee health records and personally identifiable information (PII) in multiple database locations. One IBM InfoSphereGuardium Encryption Expert customer found that its ERP implementation had sensitive data of various data types spread over 200 database columns.
InfoSphere Guardium Encryption Expert supports SAP software. It helps satisfy compliance requirements by encrypting all database files along with reports and log data. Guardium Encryption Expert encrypts databases and files in place and avoids the need to rearchitect databases, files, or storage networks. Inserted above the file system or logical volume layers, Guardium Encryption Expert is transparent to users, applications, databases, and storage subsystems. It requires no ABAP programming language coding and no modification to SAP software or the database, so deployments can be managed in weeks rather than months. These benefits are included:

- Protecting data at the source: At the data storage level (not at the application level or access layer level)
- Centralized key management, policy management, and audit capabilities
- Strong separation of duties and compliance ready
- Minimal performance impact to databases and applications
- Support for all data types and index types
- Installs quickly in a matter of days and is completely transparent to applications
- Easily extensible protection to log files, configuration files, and other database output
- SAP certified

Integration scenario: Protecting an outsourced SAP installation

Let us use an example situation in which an organization has outsourced its SAP installation, but needs to ensure that the outsourcer will not be able to read or tamper with SAP data stored on the outsourcer's premises. Such a requirement is, for example, imposed by German protection laws for especially sensitive data.

IBM solutions can protect the organization on three different levels:

- File system: By using Guardium Encryption Expert, access to the database files directly through the file system is prevented, even root access.
- Database: By using Guardium Database Activity Monitor, all access to the database via its native SQL interface is being monitored and subjected to fine-grained security policies.
- Application: By using best-practices within the SAP rights management, the SAP application itself is protected against unauthorized access.

More information: More information about InfoSphere Guardium Encryption Expert can be found at the product website:

http://www.ibm.com/software/data/guardium/encryption-expert/
IBM InfoSphere Optim

IBM InfoSphere Optim™ enterprise data management solutions focus on critical business issues, such as data growth management, data privacy compliance, test data management, e-discovery, application upgrades, migrations, and retirements. Optim aligns application data management with business objectives to help optimize performance, mitigate risk, and control costs, while delivering capabilities that scale across enterprise applications, databases, and platforms.

Integration with SAP applications

Organizations rely on critical SAP applications to support daily business operations, so it is essential to ensure privacy and protect application data no matter where it resides. However, the same methods that protect data in production environments might not meet the unique requirements for non-production (development, testing, and training) environments. How can IT organizations protect personal data, including employee and customer information, in addition to corporate confidential data and intellectual property? Industry analysts recommend “de-identifying” or masking data as a best practice for protecting privacy. But what are some of the requirements for selecting a data privacy solution?

The ideal data privacy solution must provide the necessary data masking techniques to satisfy both the simplest and most complex privacy requirements. Masking techniques must also produce results that reflect the application logic and preserve the integrity of the data. To help support your data privacy compliance requirements, Optim provides comprehensive data masking techniques.

- Application-aware masking capabilities help ensure that masked data, like names and street addresses, resembles the look and feel of the original information.
- Context-aware, prepackaged data masking routines make it easy to de-identify data elements, such as social security numbers, payroll data, and email addresses.
- Persistent masking capabilities propagate masked replacement values consistently across applications, databases, operating systems, and hardware platforms.

IBM InfoSphere Optim Solutions for SAP Applications can streamline testing and development cycles, reduce storage requirements, and improve test coverage and accuracy for your SAP application life cycle events. With Optim, organizations can de-identify data in a way that is valid for use in development, testing, and training environments, while protecting data privacy.
InfoSphere Optim Test Data Management Solution for SAP applications

With a familiar point-and-click SAP graphical user interface (SAP GUI), InfoSphere Optim Test Data Management reduces the preparation time, expense, and manual effort required to create manageable real-world data scenarios. Leveraging pre-built business objects, InfoSphere Optim can extract discrete subsets of production data based on user-defined criteria and copy it to any other system, accurately capturing the SAP data that you need for testing.

InfoSphere Optim Test Data Management Solution for SAP Applications includes out-of-the-box masking capabilities to protect sensitive data from misuse.

This is what you can do with InfoSphere Optim:
- Apply a range of pre-defined or custom masking techniques for SAP applications to protect sensitive data in SAP instances for testing, training, and development.
- Leverage out-of-the-box libraries to mask data with realistic, but fictional, values to speed testing cycles while protecting confidential data from unintentional disclosure.
- Implement masking techniques, developed and implemented directly in SAP applications via ABAP to accelerate needed testing scenarios and deployments.

The InfoSphere Optim out-of-the-box masking policy libraries include this information:
- First and last name
- Address
- Person
- Company name
- National ID

InfoSphere Optim System Analyzer for SAP applications

When businesses approach any SAP application update or modification, they must understand how those changes will impact their existing environment. For example, if they install a support pack, will it affect customizations that are already implemented? If they are adding an enhancement pack, do they need to ensure that other SAP systems can accommodate the enhancements?

IBM InfoSphere Optim System Analyzer for SAP Applications is a diagnostic tool that automatically identifies SAP system changes and helps SAP teams understand the impact of those changes on the entire SAP environment. This SAP-certified, web-based application combines a powerful analytical engine with approximately 200 prebuilt templates to support major life cycle events, such as
the deployment of application upgrades and support packs, and to provide SAP sites with the needed insight to ensure a smooth application delivery.

Leveraging a drag-and-drop workflow, templates can be easily customized to support the unique needs of each organization’s SAP landscape. Administrators can extend the support pack templates to identify which test cases should be run when implementing a particular support pack. The templates also can be extended to show gaps in test cases or to highlight affected transactions for which there is no test case.

InfoSphere Optim System Analyzer automatically identifies SAP system changes for these key application life cycle events and can provide automated, in-depth analysis for multiple systems and applications. By analyzing the before and after images of the data, it documents the SAP baseline and automatically detects any differences, providing a comparison of data to authenticate test completeness. Results are presented in a concise report, saving administrators countless hours of manual inspection.

**InfoSphere Optim Business Process Analyzer for SAP applications**

InfoSphere Optim Business Process Analyzer for SAP Applications, used in conjunction with InfoSphere Optim System Analyzer, establishes the traceability between the data structure changes and your SAP business processes, identifying how they impact one another. InfoSphere Optim Business Process Analyzer automatically captures the SAP business process from your SAP system and then provides a graphical view of how changes will impact the business process.

**More information:** For more information about InfoSphere Optim Solutions for SAP Applications see this URL:

http://www.ibm.com/software/data/optim/sap/

**IBM WebSphere DataPower**

IBM WebSphere DataPower SOA Appliances are purpose-built network devices that offer a wide variety of functionality, such as the securing and management of SOA Applications, enterprise service bus integration, and high-speed XSL execution. A hardened appliance, DataPower provides robust security features, including tamper protection of the device itself.
The following primary services are provided by the DataPower devices:

- Multi-protocol gateway
- Web Service Proxy
- XML firewall
- Web application firewall
- XSL Accelerator (Proxy)

More information: For more information of the product, consult this website:
http://www-01.ibm.com/software/integration/datapower/

Integration with SAP applications
DataPower provides many alternatives for integration of SAP systems and applications into a SOA or federated security scenario. Examples are the use of authentication proxy or enterprise service bus. Because DataPower and SAP solutions both support the SAML standard, they can be used for such integration scenarios accordingly. More integration information about DataPower solutions can be found in 12.5, “Service-based single sign-on to SAP backend systems using Federated Identity Manager and SAML” on page 341, and 12.6, “Integrate SAP into SOA by federating the SAP login ticket” on page 343.

By configuring DataPower and SAP solutions to support Sender Vouches SAML Assertions, you can allow single sign-on to an SAP NetWeaver Application Server ABAP. A brief overview of this can be found in “Sender-Vouches subject confirmation method” on page 50. The following requirements have to be met to enable such a scenario.

These are the SAP NetWeaver environment requirements:

- AS ABAP must be Version 7.00 SP14 or later.
- SAP Cryptographic Library 1.555.24 or later must be installed.
- SAP Notes 1254821 and 1325457 must be implemented in the environment.
- SAML Sender-vouches is supported with releases AS ABAP 7.00 (SP 15) and later. Ensure that the corresponding SAP Notes have been applied.

The mutual requirements are that a shared certificate must be created and installed on both SAP NetWeaver and DataPower. Many third-party tools can be used to generate this certificate, including DataPower’s cryptography toolset. Using DataPower is preferred, as it automatically installs the certificate and key after generation.
IBM System z and mainframe security

IBM System z represents the IBM mainframe computer series. These computer systems are unique for their performance and availability, and accordingly the \textit{z} stands for \textit{zero downtime}. The systems are built with spare components capable of hot failovers to ensure continuous operations.

IBM System z offers a number of differentiating benefits that can have a significant impact on those businesses that rely on SAP applications.

The unique strengths that System z brings to your business include these:

\begin{itemize}
  \item Near-continuous availability and disaster recovery
  \item Reduced infrastructure complexity and improved operational efficiency
  \item Scalability to grow, eliminating the need to partition data across servers
  \item Comprehensive protection of critical data from security threats
\end{itemize}

Along with the ability to deliver near continuous availability, IBM System z and IBM DB2 for z/OS offer substantial scalability to support high throughput for SAP databases. The result is an enterprise-level industrial-strength design that can reduce the need to separate different SAP database servers onto different hardware. This capability leads to fewer complications associated with managing large numbers of servers, which frequently result in data consistency concerns, and complex system management and integration issues.

\textbf{System z and SAP security}

Data and business information, which organizations collect and store, are extremely valuable. This might be sensitive information about customers and business practices or data that keeps organizations running every day. If something happens to that data the impact can become critical to the business. Therefore, organizations need to make sure that data is protected on the network or storage devices and that only authorized people have access to that data.

IBM offers well-known solutions in this context. System z has an integrated and complete security architecture that goes all the way back to the S/360 architecture. The modern System z family of servers provides data encryption at
all storage levels, secured key and storage management and hardware assisted communication encryption. The seamless integration of SAP security with z/OS and DB2, sharing common security directories via LDAP, allows access only for authorized users, in adherence to their individual rights and roles.

System z servers are currently the only commercially available servers that have been certified for the security of their logical partitions at Evaluation Assurance Level 5 (EAL5) of the Common Criteria Security Certification international standard. This certificate demonstrates that System z can be used in environments where strict and secure separation of workloads is a key requirement. Many distributed solutions do not even come close to this set of products and system capabilities. Their data encryption is typically an add-on, and, in cases of failures, most distributed operating systems do not contain much or any recovery code. If the operating system suffers a problem, the typical response is to reboot' the system. Conversely, the IBM z/OS operating system has a recovery code called functional recovery routines (FRR) for every major routine in the system. If problems occur, the operating system has the ability to refresh the code and keep the system up and running.

IBM System z and its DB2 data-sharing capabilities allow customers to do hardware and software upgrades and maintenance without the need to stop their SAP applications. With growing demands from government regulation and industry standards, customers need to be able to report on security efforts. Systems from IBM offer a range of capabilities to keep data and applications secure and report violations.

System z in combination with z/OS and DB2 delivers an SAP platform that can run at a higher system utilization rate than many other competitive systems, without suffering response time degradation. Additional platform benefits include the extensive granularity available in the environment, and the ability to upgrade the system on demand, often within hours, and frequently without incurring application outages.

IBM System z, z/OS, and DB2 are well known for their strong security features. IBM RACF facilities in conjunction with SAP internal security make a powerful combination. When using IBM DB2 Connect™, the communication between an external SAP application server and the SAP database server is encrypted for an added level of security.

IBM also offers hardware-assisted encryption processing that is separate from the main System z processors that execute the SAP and DB2 application workload. This separation avoids consuming valuable processor resources for security processing, eliminating overhead that happens so often in other environments.
SAP on System z security best practice solutions include these:

- Common Criteria EAL 5 certification
- DB2 trusted context with SAP enhancing authentication
- DB2 database roles with SAP enhancing authorization and auditing
- Encrypting certain SAP tables in DB2 with strong encryption
- IBM DS8000® full disk encryption for DB2 and SAP strong encryption
- TS1120 tape drive encryption for DB2 and SAP strong encryption
- IBM DRDA® data stream encryption for SAP application server DB2 data traffic

The System z capabilities for security, which are derived from its architecture, can be augmented by different IBM security products:

- **IBM Tivoli Directory Server on z/OS/on Linux for System z**\(^1\)
  - Allows z/OS LDAP to be used as the central authentication repository
  - Enables SAP data sync with SAP LDAP Connector for SAP AS ABAP user repositories
  - Permanent user store for SAP AS Java user repositories (User Management Engine (UME))

- **IBM Tivoli Directory Integrator running on System z/working with data on System z**
  - Flexible data sync and real-time user data handling
  - Connectors for SAP AS ABAP user repository, Business Object Repository (BOR), ALE/IDoc

- **Tivoli Identity Manager running on System z/managing System z Security Resources**
  - SAP managed user repositories on AS ABAP and AS Java (UME)
  - SAP GRC integration for Separation of Duties validation

- **Tivoli Access Manager for e-business running on System z/securing System z access and Tivoli Federated Identity Manager on System z**
  - SAP AS Java/ABAP access control via TAMeb WebSEAL
  - Enables SAP for federated single sign-on

\(^1\) Also see 7.2, “Tivoli Directory Server on z/OS and SAP solutions” on page 149.
IBM Power Systems and AIX Security

IBM Power Systems™ is the IBM POWER® architecture-based server product line. POWER processor technology is an instruction-set architecture that spans applications from consumer electronics to supercomputers. POWER processors provide the foundation for designing workload-optimized systems in conjunction with software and expert domain knowledge. To achieve maximum performance, POWER processor-based systems are designed with workload-optimizing technologies. For example, Intelligent Threads technology dynamically switches the processor threading mode to deliver optimal performance for different workloads. TurboCore mode also offers the option to optimize the system for frequency and cache utilization, delivering the maximum per-core performance for database and transaction workloads. IBM Active Memory™ Expansion helps reduce memory costs by enabling physical memory to be logically expanded up to 100% for some workloads, such as for SAP applications.

The IBM System p® operating system can be either AIX or a 64-bit version of Linux. The AIX operating system is an open standards-based UNIX operating system for IBM UNIX OS-based servers. AIX operates on the IBM systems based on Power Architecture® technology. AIX includes a number of features designed to provide a secure IT environment. AIX allows you to perform tasks such as hardening a system, changing permissions, setting up authentication methods, and configuring the Common Criteria Security Evaluation features.

More information about SAP on System z: For more information see these websites:
- SAP on IBM System z
  http://www-03.ibm.com/systems/z/os/zos/features/sap/
- System z Solution Edition for SAP

More information about System z Security: Also see the IBM Redbooks publication Security on the IBM Mainframe, SG24-7803.
AIX security capabilities include these:

- Role-based access control (RBAC) offers simplified administration and least privilege use hardening.
- Trusted AIX/Multi Level Security (MLS)* offers labeled security and mandatory access controls.
- Enhanced long password hashing and support for pass phrases.
- AIX Stack Execution Disable (SED) prevents the successful exploitation of many types of buffer overflows.
- RealSecure Server Sensor combines several protection technologies into a single multi-layered agent that protects AIX servers and applications from known and unknown threats with an integrated firewall and a vulnerability-centric Intrusion Prevention System (IPS).

**SAP solutions on POWER**

Organizations have to rely on their IT infrastructures to support business-critical applications, like from SAP. IBM Power Systems servers have a proven architecture and offer a highly flexible, scalable, reliable, and virus-resistant platform. They can easily be managed and provide proven flexibility to quickly adapt to changing business requirements and new software paradigms, for example, when upgrading to new SAP releases, introducing Unicode data representation, or modeling business processing aligned to the SAP SOA methodology.

Unlike small, siloed Windows and Linux-based systems, IBM Power System servers allow you to focus on your business, rather than spending extensive time and money on managing server farms. IBM POWER7® and AIX continue performance and virtualization leadership at a high level with a low total cost over time. These are the primary benefits for SAP customers:

- Excellent application performance and stability.
- Throughput optimized systems by intelligent threads (SMT4) deliver nearly unlimited scalability.
- High resiliency by inherent system design (RAS) and special virtualization features.
- Much higher degree of flexibility for deployment and operations of SAP applications and instances.
- Concurrently run IBM i, AIX, and Linux (Novell Red Hat, SUSE SLES) on the same server.
- IBM PowerHA® SystemMirror (IBM HACMP™) provides active/standby datacenter and multi-site disk clustering solutions for resiliency.
All these factors are a high priority for SAP solutions, too. Hence, there is optimal synergy between IBM POWER platform, the PowerVM® virtualization layer, the AIX operating system, and the SAP KPIs for enterprise-critical systems.

In addition to the above hardware benefits, IBM Power Systems with an AIX or Linux operating system offers unmatched virtualization capabilities, as there are these:

- IBM Micro-Partitioning®, supporting the consolidation of many SAP instances by allocating down to one-tenth of a physical core.
- Multiple shared processor pools for autonomous load balancing in combination with a reduction of processor-based software license fees.
- Virtual I/O, simplifying peripheral set up around SAP servers and reducing resulting errors. It also exploits advanced native storage subsystem features by NPIV-support.
- Live Partition Mobility, reducing planned downtime by moving live SAP partitions from one box to another.
- Advanced Memory Expansion (AME), virtually extending your physical memory per LPAR/server.

The AIX operating system (SAP certified) exploits the advanced POWER hardware features and provides Workload Partitions (WPARs). WPARs offer lean and efficient OS-based virtualization for smaller SAP instances.

In sum, POWER technology addresses the major pain points of SAP solutions with unequaled flexibility in configuring their SAP landscape and allowing for quick responses to changing business demands.

**Using AIX Security Expert to harden your AIX running SAP applications**

SAP solutions are available for a broad range of platforms. This leads to the situation that SAP requires the platform vendors and customers to deal with non-SAP development-related items, such as hardening the operating system.

IBM provides a tool that helps system administrators set the AIX OS and network-layer security configuration. They designed a tool flexible enough to address different needs that different environments have, but that still provides the capability to harden the operating system up to a very high level.

The first design aspect chosen is to provide predefined security levels to give guidance. At the same time they include the possibility to configure the levels. This enables the customer to run their business at the maximum security that the application can bare. Customers often have the wrong expectation that there is a tool that they simply run and that the outcome is a pre-tested and validated
secure system, including the guarantee that the hardening will neither impact their productivity nor allow intruders to hack their system. This is not achievable. AIX Security Expert gives you a helping hand to make the best decisions based on years of experience, But at the end of the day you have to make sure that you make the correct decisions.

The second design aspect was to group the functionality. If you would like to focus only on your network, you focus only on the network-related groups.

The last aspect was the usability and manageability. To address this aspect AIX now provides a graphical interface, in addition to the command line and administrative functionality, such as re-checks, undo, re-usage of the setup via LDAP Server, and so on.

More information: For more details about the capabilities and design see the AIX Security Expert website:

Detailed information: More details about using AIX Security Expert to harden your AIX running SAP applications can be found here:
http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105143

Single sign-on for SAP NetWeaver Application Server (ABAP) on POWER Systems

Using the SAP Secure Network Communication protocol in combination with Kerberos allows single sign-on to SAP NetWeaver Application Server ABAP-based applications.

This approach makes use of the common combination of Windows workstations grouped in a Microsoft Active Directory domain and SAP applications running on Linux or AIX on IBM POWER Systems.

The integration is based on the principle that SAP offers Secure Network Communication (SNC) as a component to integrate an external security product into SAP systems. The used external security product in this context is Kerberos.

Kerberos is a network authentication system based on a key distribution model and a central repository. The idea is that every resource authenticates itself against the Kerberos server and receives a key, also called ticket. This ticket is then used during communication with others to identify the resource as a trusted
partner and enables single sign-on, as no passwords will be requested due to the ticket exchange.

The first step for a Kerberos authenticated communication is that a resource authenticates itself against a Kerberos authentication server and requests a ticket. This first ticket is the Ticket Granting Ticket (TGT) that is used for the communication with the Kerberos Key Distribution Center (KDC). Furthermore, during the first communication with the authentication server, a session key is created that can be used for encrypted communication. The Kerberos authentication server and the KDC are two independent servers and could work on different servers, but usually they are part of the same system and are seen as a unit. When the resource now tries to connect to another party, it needs another ticket. Therefore, it sends the TGT and a ticket request to the KDC that creates tickets for the communication with other Kerberos-enabled resources. The KDC will respond with a ticket for the communication between the two partners. The first resource forwards the ticket to the host with which it wants to communicate. If the host accepts the ticket as valid, the resource has been authorized to connect to the system for the duration of the ticket. The tickets have a limited lifetime and must be renewed after a while, normally after eight to ten hours. Another security setting is that every ticket is also bound to a unique Service Principal Name (SPN) that clearly identifies a user or resource. The ticket can only be used by the partner to which it was granted.

In the SAP context, the involved parties are the user on his local workstation, the SAP system on a Linux or AIX host, and the central active directory that serves as a Kerberos server, which contains both the authentication server and the KDC.

The user on his workstation that is part of a Microsoft Active directory domain is automatically authenticated against the central Kerberos server during logon. Apart from that only a Generic Security Services Application Programming Interface (GSSAPI) library must be installed on the workstation, as SNC uses the GSS-API interface to communicate with Kerberos. The same is true for the Linux/AIX host. It also needs the GSS-API lib.

Furthermore, the host that the SAP systems runs on must be registered as the domain user. Therefore, a computer account is created in the active directory user section. As there will not be a physical user on the host that will actively authenticate it against the Kerberos server (the SAP system will be started once and then run on its own without direct interaction by an administrator), we will use another mechanism.

On the active directory server, a keytab can be created that holds the credentials of the host and its encrypted password of the computer account. This keytab is then copied to the host. It can be called by a local user, and all applications running under this user will be enabled for Kerberos authentication. As the SAP system is running under a user ID called <sid>adm, this user must be authorized
to call the keytab. The host (not the user himself) is then authenticated against the Kerberos server. The user, however, will hold the ticket needed for communication with other parties. So, if there are applications on one host started by different users, there must be one ticket for each user. The tickets must be renewed at regular intervals. This can be done in an automated job.

After the SAP system host and user have authenticated themselves against the Kerberos server, the user is able log in to the SAP application without password input. The authentication is done in the background.

Now you need to perform these steps to enable your own SAP system for SSO:

1. Configure the Kerberos Client.
2. Add the Linux/AIX Server as the host to Active Directory and create keytab.
3. Enable the SAP NetWeaver Application Server (ABAP) for SNC.
4. Enable the SAP User for SNC.
5. Enable Kerberos and SNC on the Windows Client.
6. Test SSO.

**More details:** For a detailed description of this scenario see the paper *Single Sign On for SAP NetWeaver Application Server (ABAP) on Power Systems*:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101578

---

**Conclusion**

This concludes the introduction of the IBM Security offerings that provide integration capabilities with SAP solutions beyond the IBM Security identity and access management portfolio.
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Identity Management Design Guide with IBM Tivoli Identity Manager*, SG24-6996
- *Centrally Managing and Auditing Privileged User Identities by Using the IBM Integration Services for Privileged Identity Management*, REDP-4660
- *Propagating Identity in SOA with Tivoli Federated Identity Manager*, REDP-4354
- *Federated Identity and Trust Management*, REDP-3678
- *DataPower Architectural Design Patterns: Integrating and Securing Services Across Domains*, SG24-7620
- *Security on the IBM Mainframe*, SG24-7803
- *IBM Tivoli Directory Server for z/OS*, SG24-7849
You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- SAP Service Marketplace
  SAP created and maintains the SAP Service Marketplace websites, including all related websites and other websites added to the SAP Service Marketplace from time to time (collectively SMP) to provide a repository where customers and SAP partners can obtain information about and access to programs and other materials made available by SAP. Such information about and access to programs and other materials might include, but is not limited to, articles, data, code, text, SAP software or related documentation, documentation and product specifications, application program interface specifications, concepts, designs, programming techniques and programming concepts, flow charts, graphics, images, training, and other services, in addition to marketing material around SAP’s products and services.

  https://websmp201.sap-ag.de

- IBM product manuals for the security-related offerings in this IBM Redbooks publication can be found at the following locations:
  - IBM Tivoli Access Manager for e-business Version 6.1.1 Information Center
  - IBM Tivoli Access Manager for Enterprise Single Sign-On Version 8.1 Information Center
  - IBM Tivoli Federated Identity Manager Version 6.2.2 Information Center
  - IBM Tivoli Identity Manager Version 5.1 Information Center
– IBM Tivoli Directory Server, Version 6.3 Information Center
  

– IBM Tivoli Directory Integrator Version 7.1 Information Center
  

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services
Integrating IBM Security and SAP Solutions
Integrating IBM Security and SAP Solutions

SAP business solutions, security, and the user and role management concepts

Many large and medium-sized organizations have made strategic investments in the SAP NetWeaver product suite as their primary application platform. In fact, SAP software is used to manage many core business processes and data. As a result, it is critical for all organizations to manage the lifecycle of user access to the SAP applications while adhering to security and risk compliance requirements.

IBM Security identity and access management integration

In this IBM Redbooks publication, we discuss the integration points into SAP that are supported by the IBM Security access and identity management product capabilities. IBM Security software offers a range of identity management (IdM) adapters and access management components for SAP that are available with IBM Tivoli Identity Manager, IBM Tivoli Directory Integrator, IBM Tivoli Directory Server, IBM Access Manager for e-business, IBM Tivoli Access Manager for Enterprise Single Sign-On, and IBM Tivoli Federated Identity Manager.

Use cases and best practices

This IBM Redbooks publication is a valuable resource for security officers, consultants, administrators, and architects who want to understand and implement an identity management solution for an SAP environment.

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks