INTRODUCTION

As the recognized leader in the Secure Identity Management space, Novell has been providing customers with the leading Identity Management solution for a number of years, from DirXML® through to the current Novell® Nsure™ Identity Manager 2.0.

One of the key aspects of Identity Management is provisioning; this includes implementations such as zero-day start and zero-day stop, as well as policy-based resource access. Nsure Identity Manager provides industry leading rule-based provisioning, as well as manual provisioning facilities through the Manual Task driver.

This paper describes how Identity Manager provisioning can be extended to sophisticated multi-step, branching workflows using the facilities in Novell’s exteNd™ suite. The paper covers:

- Novell SIM Architecture and Automated Provisioning overview
- Nsure Identity Manager Architecture summary
- Provisioning with Nsure and exteNd
- exteNd Director™ and exteNd Composer™ in the Provisioning Context
- Provisioning Workflow sample project

This paper and sample project continues the work begun with the DirXML Workflow driver (now the Identity Manager Manual Task driver) to realize Novell’s longer term vision to provide sophisticated role-based provisioning workflow for the enterprise. Novell expects to complete a major milestone in this vision with the release of a powerful purpose-built provisioning workflow environment in Q3/Q4 of this year.
NOVELL SIM ARCHITECTURE OVERVIEW

Secure Identity Management provides the foundation for secure administration and application of Identity across the enterprise. Novell SIM technologies enable robust access control and scalable administration through implementation of secure, efficiency-oriented management mechanisms and Integrated Identity.

The Novell comprehensive suite of SIM technologies includes additional SIM components enabling the creation of powerful solutions including Self Service Password & Identity Management, Web Access Control, Single Sign-On and Secure Logging & Auditing. Due to the strength of the Novell SIM technologies and their unique flexibility and advanced customization facilities, many additional solutions are possible, but here we’ll explore only those solutions which are directly relevant to provisioning.

Policy-based Provisioning

 Corporations need to deal on a day-to-day basis with the administrative tasks associated with large, dynamically changing user populations, frequent organizational changes, mergers, acquisitions and evolving extranet partnerships. As a user’s responsibilities change, so do their requirements for resource access, so existing privileges must be revoked in favor of privileges relevant to new responsibilities.

Policy-based provisioning provides a solution to the administrative problems caused by frequent workforce changes by combining the end user self-service components of Secure Identity Management with Policy-based synchronization of user accounts and passwords across the broad myriad of enterprise platforms and applications to ensure that end users have timely access to the resources they require. Furthermore, as employee, partner, customer and supplier access is no longer appropriate, policy-based provisioning enables fast, easy deactivation of privileges to ensure the continued security of information assets.
For corporations experiencing rapid workforce change, provisioning provides solutions for resource access management that scalably enable IT personnel to keep up with the never-ending need to manage a multitude of privileges while simultaneously delivering consistently high levels of customer service.

**Provisioning with Nsure Identity Manager**

Nsure Identity Manager is the Novell solution for enabling shared Identity across disparate systems and defining attribute-level authoritative sources—the critical feature required to effectively enable Integrated Identity. Identity Manager’s capabilities are provided in real time, permitting changes in individual Identity applications to be quickly propagated to the Integrated Identity and, from there, into other Identity applications throughout the enterprise as needed. The Identity Manager architecture is uniquely capable of facilitating Secure Identity Management and Provisioning by providing complex data transformation services and flexible application specific rules that expedite systems integration, including the ability to transform application specific actions and states into generalized events and Triggers. Identity Manager includes drivers (sometimes called “agents” or “connectors”) that support a wide variety of directory services, databases, computing platforms and corporate applications.

In many cases provisioning will require some action on the user’s part. The Identity Manager Manual Task Driver is designed to notify one or more users that a data event has occurred and in some cases that action is required on the users’ part. In an employee provisioning scenario, the data event might be the creation of a new User object and the user action might include assigning a room number by entering data into Novell eDirectory™ or by entering data in an application. Other scenarios include notifying an administrator that a new user object has been created, notifying an administrator that a user has changed data on an object, etc.
For the Foundation for Secure Identity Management Solutions Architectural Guide, see

For the Manual Task Driver documentation, see
PROVISIONING WORKFLOW

In many cases more sophisticated workflows than those supported by the Manual Task Driver will be required.

The Workflow Management Coalition defines workflow as the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

Novell exteNd Director provides a full-featured, robust workflow platform that was designed with just these requirements in mind. It supports sophisticated multiple step workflows with conditional logic through an intuitive visual workflow and forms design environment:

**Directory Integration** The user-related services of exteNd Director provide straightforward access to identity information for authorization, authentication, and provisioning requirements (see User Related services below for further information).

**email Notification** exteNd Director Workflow supports email notification of workflow status and action requirements through standard JavaMail* facilities.

**Intuitive design and development environment** exteNd Director Workflow provides an intuitive workflow process designer. The Workflow and Forms Designer allows you to quickly and visually create workflow processes and forms, bringing the power of visual design to workflow development.

**Workflow administration console** exteNd Director Workflow includes a user interface (UI) for managing the workflow engine, workflow queues, and workflow processes that makes it exceptionally easy to administer workflow for your business.

**Sample processes and portlets** exteNd Director Workflow includes a library of sample processes and portlets, providing an excellent starting point for building workflow applications. This paper describes a sample provisioning workflow.

**Web Services support** exteNd Director Workflow allows you to assign an activity in your workflow directly to a Web Service. Both .NET and Java*-based Web Services are supported.

**J2EE* compatibility** exteNd Director Workflow is fully J2EE compatible. This ensures that your workflow applications receive all of the benefits provided by J2EE. Your workflow-enhanced applications will be portable, scalable, and secure—and you avoid vendor lock-in by having the power to deploy your applications to the application server of your choice.

**Flexible and extensible architecture** exteNd Director Workflow provides an architecture that is both flexible and extensible. All exteNd Director Workflow functions are fully extensible and available to developers via an open Java API (Application Programming Interface) and a JSP (JavaServer* Pages) tag library.

**Identity Manager Integration** exteNd Director Workflow can be incorporated in Identity Manager provisioning through a wide variety of trigger mechanisms. Using the Identity Manager Subscriber and Publisher conventions, a partial list would be as follows:
For further information on exteNd Director Workflow, see the Conceptual Overview later in this paper.

**Novell exteNd Overview**

exteNd Director Workflow is part of the Novell exteNd suite of products. exteNd provides intuitive visual tools, wizards and services that assist the corporate application developer in rapidly assembling and deploying vital business services such as workflow. exteNd Director provides the user interaction layer, while exteNd Composer provides the back-end integration layer, facilitating the integration of identity and transaction information which is outside the core corporate identity stores.

**exteNd Director**

exteNd Director is a very rich environment which includes facilities like Content Management and a Rules Engine; here we’ll concentrate on exteNd Director out-of-the-box facilities which directly support rapid implementation of manual provisioning workflows. These include:

- Express Portal
- User Related Services
- Rules Engine
- Visual Workflow and Forms Design

**Express Portal**

Express portal is just what its name implies: a complete, pre-configured portal application that helps you rapidly deploy sophisticated portal solutions. It provides a Web site that consolidates and organizes the information and applications your employees need, including provisioning workflow.

Express portal does the following:

- Provides authentication of portal visitors and single sign-on across multiple applications, simplifying secured access to critical business applications and information.
• Allows portal users to customize the layout of their portal pages so that information and applications are organized for easiest use.
• Allows the portal administrator to customize site-wide content and layout for common applications, groups of users or on a per-user basis.
• Includes out-of-the-box portlets for a wide variety of applications such as Novell GroupWise®, Lotus Notes®, Microsoft® Exchange, syndicated news and specific industry applications.

User Related Services
The user-related services of exteNd Director provide straightforward access to identity information for authorization, authentication, and provisioning requirements.

exteNd Director is made up of a variety of subsystems, each of which provides a group of services.

The exteNd Director user-related services are provided by three of the subsystems:
• The Directory subsystem
• The Security subsystem
• The User subsystem

The Directory subsystem provides services for authenticating users to the underlying application server, and enables you to manage user and group repositories using the application server’s own or back-end directory servers such as LDAP (including Novell eDirectory), NT Domain, and NIS+.

The Security subsystem provides role-based security services to restrict user access to portal pages and workflows, and it provides ACL-based security services to restrict access to subsystem functionality including specific provisioning entitlements. You can define security roles and access controls lists (ACLs) programmatically or interactively using the Director Administration Console (DAC), the exteNd Director Web-based administration user interface, to authorize users to perform certain operations.

For further information on user-related services (including the user subsystem), see http://www.novell.com/products/extend/pdfs/director40_user_related_services.pdf

Rules Engine
The exteNd Director Rules Engine is callable from Workflow activities; it allows the encapsulation of business logic into simple rules which can be maintained by business users through a wizard-driven environment. This makes it easy for business users to modify the users that an activity applies to, or change email addressing on a workflow notification.

Workflow Conceptual Overview
Like all workflow systems, exteNd Director Workflow is process oriented. It is based on a process definition that is a representation of what the workflow does and how it does it. A process definition includes common workflow concepts such as activities (tasks), links (procedural rules), and workitems (documents and information). A process definition is used by the workflow engine to execute a workflow at runtime.
Activities Activities represent the things that may be done in a workflow. In a provisioning process, for example, activities would include requesting access to a resource and approving access to a resource. exteNd Director Workflow includes standard activity types for the most common activities. These include activities that represent user interactions, automated processes, Web Service calls, synchronized activity merges, and process startup or completion. You can create custom activities when you need to perform other types of tasks.

Links Links represent the decision points in a workflow process. Which activities are done, who does them, and their sequence are defined by links. A link specifies the source activity, the destination activity and addressee, and the conditions for which the link is valid. exteNd Director Workflow includes standard link types for the most common behaviors including conditional links, logical links, and business rule links. You can create custom links when you need to.

Workitems and documents Workitems represent the state of a process and act as containers for documents and properties. They are routed to activities and addressed to users according to the process routing logic. Workitems can contain three types of document: DOM (Document Object Model), application-specific document identifiers, or an URL to a document that is stored elsewhere (such as in the exteNd Director Content Management subsystem). Properties are name/value pairs that are stored with a workitem.

textDirector includes an intuitive visual designer for creating and editing workflow process definitions. The Workflow Designer includes graphic tools that allow you to professionally lay out, annotate, and format workflow process definitions.

Using the Workflow Designer, you create a new process definition and set its properties, including the process name and the roles that are allowed to create workitems for the process.
Workflow Administration
The Workflow subsystem includes UI components for administering workflow, including the Engine and Queue Administration Console and the Process Administration Console.

The Engine and Queue Administration Console provides the ability to start, suspend, or shut down the workflow engine and queue.

The Process Administration Console allows you to manipulate the execution of a process instance by suspending or resuming a process or the activities of the process. The Process Administration Console also allows you to list and see the status of processes and activities.

For further information on Workflow, see http://www.novell.com/products/extend/pdfs/director40_workflow.pdf

Visual Form Design Environment
Using innovative XForm technology, the Novell exteNd Form Designer reduces the time necessary to create user interfaces for provisioning workflows by automatically generating user interfaces from Web Service meta data. These user interfaces can then be securely accessed via a standard Internet browser.

The Form Designer contains the following features:

- Visual editing and positioning of user interface controls
- Drag-and-drop data binding of controls to instance data
- Visual event editor to visually create/edit event handlers on XForms controls
- Instance data editor—You can supply production data in the development environment to test your form's runtime characteristics
• XPath Navigator
• Visual Cascading Style Sheet (CSS) editor
• Real-time Form preview—See the look and feel of your Web form in the development tool
• Zero-admin Form deploy—Form modifications are dynamically reflected on the server without explicit intervention

Page Flow Modeler
Few provisioning workflow activities consist of a single form. Therefore, Novell exteNd provides a complete visual-design tool, called the Pageflow Modeler to quickly assemble page forms and create flexible applications that can execute either as standalone objects or as part of a Portlet Specification 1.0-compliant portal.

• Productive graphical tools—These tools allow you to annotate, format and assemble forms into a page flow
• WebService Wizard—A wizard interface for the creation of portlets that consume Web Services
• Composer Pageflow—A wizard that allows you to quickly build a portlet based on a Novell exteNd Composer service
• Database Pageflow—A set of visual tools that allows users to easily create flows that manipulate database records
• ScopedPaths—An “XPath”-like syntax for accessing various types of information. This information should reduce the amount of custom Java activities required for simplistic data movement
• SmartLinking—A mechanism that reduces the number of links that must be defined to navigate from one activity to another
• XForms—Links are dynamically mapped by the flow, based on names and functions
• Universal Link—A UI mechanism for ordering the evaluation of outbound links in an XForm
• Checkpoint Activity—A specialized flow activity encapsulates the complex task of Web application transaction management
• Form Logging—A mechanism that logs the inner calls made during the processing of a page flow

In addition to presenting pages for user interaction, page flows can perform background-processing tasks. For example, a page flow might invoke a Web Service or access a directory.

exteNd Composer
While many enterprises struggle with highly complex IT infrastructures, a few have created environments in which people communicate and collaborate easily and always have the tools and information required to work effectively. Two characteristics distinguish these highly successful enterprises from the rest: agility and adaptability.

exteNd Composer enables organizations to non-invasively, Web Service-enable LDAP directories, zSeries* (OS/390*) mainframes, iSeries* (AS/400s*), legacy VAX/VMS’s, packaged vendor and UNIX* applications and numerous others systems. In addition, exteNd Composer includes a Web Services-based process manager that supports complex process automation and modeling including timeouts and retries.

exteNd Composer Web Services can be seamlessly incorporated in exteNd Director Workflows using the included wizards, extending secure identity based provisioning of resources across the enterprise. Amongst the many connectors which exteNd Composer provides is LDAP Connect:

exteNd Composer LDAP Connect
The exteNd Composer LDAP Connect lets you build components and services that are directory-aware. Your component or service acquires the power to act as an LDAP client. It can make queries against (or even update the contents of) any directory—regardless of vendor—that supports the LDAP protocol.

With the aid of the LDAP Connect and Composer, you can build “directory awareness” into your XML integration applications (whether they’re Web Services or private apps running in a local context). Your
LDAP-aware app can push data into or pull data from any LDAP-accessible data store, using XML as the interchange format. (DSML is the XML dialect that is actually used.) And you can do this without having to know anything about DSML. For example, you can write a component (perhaps part of a larger web service) that retrieves the phone number, e-mail address, and title of a company employee from a company directory; or updates a directory object or attribute once a manual provisioning workflow is complete in the same way as an Nsure Identity Manager publisher channel. If the information your app needs resides in two or more directories, you can merge the information from separate directories before displaying it to the user or passing it to another component in your application.

The key to the power and flexibility of the exteNd Composer LDAP Connect is its ability to work with DSML (Directory Services Markup Language), which is an industry-standard XML grammar for encoding directory requests and responses. (See the more detailed discussion further below.) Since DSML is just a dialect of XML, it shares all of XML's advantages in terms of being human-readable, machine-parsable, transportable, firewall-friendly, etc. The data in a DSML document is easily accessed, transformed, and repurposed.

You don’t even need to create, or keep on hand, actual DSML documents in order to work with the LDAP Connect. Composer will create the necessary DSML for you, on the fly.

For further information on exteNd Composer, see

For information on the exteNd Composer Business Process Modeler, see

For further information on the exteNd Composer LDAP Connect, see

SAMPLE PROVISIONING WORKFLOW

Sizing
At the current time exteNd Director Workflow is best suited to single-CPU deployments with a moderate number of workflow activities on a queue.

These scalability constraints are being addressed in exteNd 5.2, expected in the late Q1 or early Q2 2004 timeframe. In the interim, exteNd Composer Business Process Manager based workflows provide greater scalability and support multiple-CPU clustering.

Summary
This document will walk through a step by step procedure to setup a sample workflow using exteNd Director, Identity Manager 2.0, and Novell eDirectory. For this example we will use the default project included with the exteNd suite 5.0, and the demo files included.
Configure the security realm to use eDirectory

Adding the AUX Class to eDirectory

Before you deploy a project that implements one of the LDAP realms, you need to add a UUID attribute that allows the Director and User subsystem APIs to access the LDAP realm. You import this class using the NDS® Import Wizard in the Novell ConsoleOne® eDirectory tool.

Importing the UUID auxiliary class in ConsoleOne:

- With the NDS container selected in ConsoleOne, select Wizards>NDS Import/Export.
- Click Import LDIF File and choose Next.
- Navigate to the ldif file in your Director installation path and select it, the default location is “C:\Program Files\Novell\exteNd5\Director\bin\extElemImport.ldif”. Click Next.
- Verify the LDAP host name and port, choose Authenticated Login, and specify your administrator DN (distinguished name) and password.
- Verify the information and click Finish.

Configuration of the exteNd Application Server

How to configure the exteNd App Server LDAP Realm

- Launch the Server Management Console - this can be found under the Novell exteNd 5.0 | Application Server program group.
- Add a security provider:
  - go to the security tab
  - click on security provider
  - chose LDAP
  - click add, you will see the window pops out and fill in your ldap server info.
  - first screen: select default
  - second screen:
    - Under server type in your ldap server name
    - User Login Attribute: “cn”
    - User Name: your admin user full “dn” eg: cn=admin,o=novell
    - Password: admin user password
    - leave other options as default
  - third screen:
    - Group Location: your group location
    - Group Filter: “(objectClass=group)”
    - Group/User Attribute: “uniquemembership”
    - leave other options as default
  - Fourth screen:
    - user location: your user location
    - user Filter: “(objectClass=person)”
    - leave other options as default , click finish
• back to general tab
  o select “LDAP” as Default Security Realm
  o chose the ldap server you just configured as Default Security Authority
• Click Update

**Configuration of the Director Project**

*How to configure Director to talk to eDirectory*

• Open Director Designer
  o this can be found under the Novell exteNd 5.0 | Director program group
• Open your project
  o If you are working with the default project this should open automatically - or simply browse by using File | Open Project
• Once the project is open select the Project Menu | Director | Configuration
• Click on Directory tab
  o change the realm to "extend Server LDAP"
  o At the bottom of the Directory tab there is addition tabs - select the “Directory Ldap Options” tab and change the following:
    o Administrator: your ldap server admin user full dn
    o Password: admin user password
    o Ldap host: your ldap server host name and port
    o new user container: the container name you want the new user to be created from, usually I put my root container name there
    o User Container DN: the root container of your user group anthor: the root container of your group
    o Root Container distinguished Name: the root container for your container

• Click on User tab:
  o select “Directory Ldap realm “ to the Data Store
  o click ok.
• Rebuild the project and deploy
  o Select the Project Menu and select Rebuild All
  o Select the Project Menu and Deploy Archive

*Configure user attributes that will be used by workflow.*

The following procedure will demonstrate how to configure some sample attributes to use with workflow. By default Director can query any single valued user attribute using LDAP. We will create three Boolean attributes for this example within NDS. This example will assume that these attributes are automatically created when the user is provisioned to eDirectory.
Create custom attributes in ConsoleOne:

- Open ConsoleOne, and open choose Schema Manager from the tools menu.
- Choose the Attributes tab and click on “Create” to start the wizard.
- Give a name of Oracle* for the “Attribute name:” and choose next.
- Choose Boolean for the Syntax from the dropdown list.
- Click next, then finish.
- Repeat these steps, and create the attributes MySQL*, and Linux*.
- From Schema Manager, choose the Classes tab.
- Find the User class, and click on “Info”.
- Click on add attribute, and select the three attributes Oracle, MySQL, and Linux.
- Choose “Ok”, and “Close”.
- Choose properties of a sample user, and click on the Other tab.
- Choose “Add”, and add the attributes: Oracle, MySQL, and Linux with a value of false.

Create LDAP attribute mappings in ConsoleOne:

- Select the LDAP Group object in ConsoleOne and choose properties.
- Choose the attribute mapping tab, click add to create a new LDAP mapping. From the dropdown list select your NDS attribute, and assign a unique name in the primary LDAP attribute field. For this example you will need to create an attribute mapping for manager, and the 3 custom attributes with a Boolean value.
- Click “ok”, and refresh the LDAP server by viewing properties the LDAP Server object in ConsoleOne and choose “Refresh LDAP Server”.

-
Add the attributes to the Director LDAP realm:

- Open exteNd Director Designer, and open your project. If you are using the default project, this should open automatically when you start Director Designer.
- Choose Project | Director | Configuration from the menu to open the project configuration page.
- Choose the User Tab to define the attributes.
- Add Oracle, MySQL, and Linux attributes that you created in eDirectory to the LDAP group object using the LDAP name that you gave it earlier. You may separate multiple attributes using a comma in this list.
- Add the attribute “manager” to this list also. The manager attribute will be used to determine the direction of flow in this project.

Deploy the DEMO project files

Demo.zip instructions

- Extract the contents of the Demo.zip file to your Director Project. The following directories will be built:

  /DEMO/Documents
  /DEMO/DemoDirector
  /DEMO/Setup
• The Document Directory holds the documents required to copy to the CM subsystem for this application if you want to use default documents stored in the CM subsystem. It also holds all the documentation for this framework.
• The DemoDirector directory holds all the artifacts for the Resource Request application. This is the subproject that will be added to your Director project.
• The Setup directory holds the Setup.bat file that will automatically update the project SPF, web.xml and resourceset.xml files to incorporate the application into your Director project with vulturing capabilities.

Setup.bat instructions

• From the Setup directory, run the setup.bat program. It will start a java program to help you setup the Resource Request application in your Director Project.
• You can either accept the default of: “C:\Program Files\Novell\extend5\Projects\ExpressPortal\ExpressPortal.spf” by pressing the Enter key or you can type in the full path name of your Director project.
• Type ‘Y’ to modify the Project SPF file. This will insert the DemoDirector.spf file as a subproject to your Director project. It will also set all the necessary links to the content of the Resource Request application.
• Type ‘Y’ to modify the web.xml file. This will add the taglib entries into this file for the JSTL tag libs.
• Type ‘Y’ to modify the resourceset.xml file. This will set the DemoDirector subproject to make use of vulturing.

Post Deployment

Content Management Documents

If you want this application to work with the Content Management subsystem, then you will need to copy default documents to your Content Management subsystem using WebDAV. Create a folder called ‘ResourceRequest’ and copy these files from the Document directory:

ResourceRequest.xml, Queue.xml, SearchRequest.xml, ResourceRequest.xsd

Users/Security

All queues are assigned role level securities. The following roles have been defined:

Managers: associated to eDirectory group cn=Managers,o=Novell
AppAdmin: associated to eDirectory group cn=ApplicationAdministrators,o=Novell

Using the ConsoleOne, create the following users:

• Managers:
  manager1, manager2, manager3

• Application Administrators:
  appadmin1, appadmin2, appadmin3
Users: user1, user2, user3

Using ConsoleOne, create the following groups:

- cn=Managers, o=Novell
- cn=ApplicationAdministrators, o=Novell

The following attributes must be set for all users:

Given Name, Last Name, Full Name, Email, Location, Oracle, MySQL, Linux

Shared Page

Using the administration functions create the following shareable pages to access the JSP pageflows and XFORMS pages and assign the appropriate user level security to them:

<table>
<thead>
<tr>
<th>Shared Page</th>
<th>Portlet Content</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Resource Request</td>
<td>ResourceRequest</td>
<td>Container o=Novell</td>
</tr>
<tr>
<td>Manage Resource Request</td>
<td>QueueMainMenu</td>
<td>Group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cn=Managers, o=Novell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cn=ApplicationAdministrators, o=Novell</td>
</tr>
</tbody>
</table>

Customize the Demo project file

Change the IDM2 Directory

To change the directory that Director will use to output the XML form for the delimited text driver you may change the scoped path in the workflow. The following procedure will explain how this is accomplished.

- Open Director Designer, and open the project that the DEMO files were installed to.
- Choose File | Open and browse to \Novell\exteNd5\Projects\ExpressPortal\data\workflow-process\ResourceRequest.xml
- This should open the ResourceRequest.xml in the Workflow Editor.
- Right Click and choose Properties of the “Create DirXML File” workitem
- Choose the “Copy Scoped Paths” tab, and click on edit to change the location.
  Default: /String/C:/temp  → /Application/DirXMLDirectory
- Choose File | Open and browse to \Novell\exteNd5\Projects\ExpressPortal\data\workflow-process\ApplicationAdminApproval.xml
This should open the `ApplicationAdminApproval.xml` in the Workflow Editor.

- Right Click and choose Properties of the “Create DirXML File” workitem
- Choose the “Copy Scoped Paths” tab, and click on edit to change the location.
  Default: `/String/C:/temp → /Application/DirXMLDirectory`

**Change SMTP for eMail messages**

To change the email SMTP server, use the following procedure to modify the pageflow in Director Designer.

- Open Director Designer, and open the project that the DEMO files were installed to.
- Choose File | Open and browse to `\Novell\exteNd5\Projects\ExpressPortal\data\pageflow-process\QueueJSP.xml`

This will open the `QueueJSP.xml` in the Pageflow Editor.

- Right Click and choose Properties of the link between “Process workitem” and “Validate Resource Request” (note: This is not the “error found” link that also combines these items).
- Click on “Edit Scoped Paths”, then on “/String/mail.novell.com → /Application/email-host” to change the host.
- Modify “/String/mail.novell.com” to reflect your mail server host.
- Click update to save changes.
- Choose File | Open and browse to `\Novell\exteNd5\Projects\ExpressPortal\data\pageflow-process\QueueXFORM.xml`

This will open the `QueueXFORM.xml` in the Pageflow Editor.

- Right Click and choose Properties of the link between “Process workitem” and “Validate Resource Request” (note: This is not the “error found” link that also combines these items).
- Click on “Edit Scoped Paths”, then on “/String/mail.novell.com → /Application/email-host” to change the host.
- Modify “/String/mail.novell.com” to reflect your mail server host.
- Click update to save changes.

Within the “Edit Scoped Paths” of the link between “Process workitem” and “Validate Resource Request”, you may also change the to, from, message, and subject fields. By default these items are configured to use information from the workflow, but you can specify a static value by using “/String/your-value”. (Note: This must be changed in the QueueXFORM.xml and the QueueJSP.xml pageflows):

- Email To:
  `/Flow/document/resourceRequest/WorkitemDoc/ResourceRequest/email/text() == > /Application/email-to`
- Email From:
  `/User/email == > /Application/email-from`
- Email Subject:
IDENTITY MANAGER 2 INSTALLATION

In this part of the document the installation of Identity Manager 2 (IDM2) is outlined for this solution. Identity Manager 2 provides data synchronization based on a business’s process rules. While exteNd Director is being used to provide the workflow in this example, IDM2 is being used provide two important functions:

1. Notifying newly provisioned users of the website used to request “company” resources. This occurs on an ADD event within the directory where the email address has been populated.
2. Handles processing the “approval” of resources by modifying the appropriate attributes within the directory.

The IDM2 Delimited Text driver is being used to facilitate the connection with the exteNd Director workflow in this example. For this particular example the installation will occur on a Windows* 2000 Server system with service pack 3. This part of the solution however could run on any Novell supported platform (ie: NetWare®, Sun Solaris*, Linux, Windows...etc.)

Prerequisites
Before beginning the installation of IDM2 you first need to have an installation of eDirectory v8.7.1 or higher running as well as an installation of iManager 2.0.2 or higher. iManager is the web based administration tool used to administer IDM2. These instructions were written with the assumption that eDirectory and iManager are both on a Windows 2000 server and that IDM2 is being installed onto that server.

For the installation of iManager 2.0.2 on Windows it is suggested that the installation occur on a clean install of the OS without Microsoft IIS service installed. This will force the iManager installation program to install it’s version of Apache and Tomcat during the installation. You MUST have a version of iManager installed before doing the IDM2 installation. The IDM2 installation will want to add the necessary plugins to the iManager installation to manage IDM2.

Installing Identity Manager 2
The installation of IDM2 can be started by inserting the IDM2 CD into the computer. The installation program will begin by default.
The installation program will prompt you for information as necessary. To continue the installation click “Next”.

- Read the license agreement. If it is agreed to please click “I accept”
- The next 2 screens are informational and describe what the different installation options will install. (See the image below for an example) Click “Next” after reviewing the different options.

- The installation will come to a summary screen depicting which options are to be installed. For a new installation on Windows 2000 the default options will give everything that is needed for this solution. These options should be:
  - DirXML Server
  - DirXML Web Components
  - DirXML Utilities
- The installation will then look for an installation of eDirectory and if a proper version is discovered will continue on with a dialog box prompting for which drivers to install. It is ok to leave the defaults selected. This will install the IDM2 engine, directory schema, and all of the drivers. In particular this solution utilizes the Delimited Text driver so it must be selected! Click “Next” to continue.
- A warning will appear about activation of the product. IDM2 will need to be activated within 90 days or else the solution will stop functioning! Click “Ok” to clear the warning.
• If the AD driver was selected for installation a warning about password sync will appear. Click “Ok” to continue past the warning.
• The installation will then prompt for authentication credentials for the eDirectory tree the solution will be installed into. Provide the correct credentials and select “Next”.
• The installation will prompt for additional components to install. Unselect the “eGuide” component as it is not needed for this installation. Only the following 2 components should be installed.
  o Novell iManager Plugins
  o DirXML Driver Configurations
• The plug-in installation installs the DirXML plugins into the iManager installation. The installation prompts you for authentication credentials and a port. The installation REQUIRES a secure port. Change the port to the secure port for iManager if it was changed during the iManager installation! Click “Next” to continue.
• The installation will again prompt for additional components to install. However this time there are not any components that are need for this solution. UNSELECT both options listed here:
  o Novell NSure Audit System Components
  o Application Components
• Finally a summary screen will appear outlining all of the options select for installation. Click “FINISH” to begin installing all of the components for IDM2.
• When the installation is finished it will prompt to review the readme file. Click “Close” and the installation will end and the readme will pop up. It is recommended that the server be rebooted after the installation.

Identity Manager 2 Configuration

Note: A basic understanding of IDM2 will help in understanding the following sections

Now that IDM2 has been installed it is time to import the driver that is going to be used in this solution. For this particular solution the Delimited Text driver will be utilized. The Delimited Text driver can take any a delimited text file or a properly formed XDS document as input and output a delimited text file or XDS document. The driver polls a specified directory for its input. Any time a file with the proper extension is dropped into the directory, the driver will consume it and process it according to the programmed business logic. For this solution the driver is going to receive a XDS document as input from the workflow occurring in Extend Director.

The driver is going to need to implement the following to satisfy the solution:

• On the subscriber channel:
  o Only allow ADD events when the email address and manager attributes are populated.
  o Send an email to the newly provisioned person to invite them to sign up for resources
  o Create an ADD-ASSOCIATION event to populate the association within the directory. For this solution the users dn will be used for the association value. This must be populated before the user object will be updated.
• On the publisher channel:
  o Send an email to the person notifying them of results of the workflow.
Modify the appropriate attributes with the directory. In this case a set of 4 boolean attributes used to signify resource status.

The following XML examples will be created for Oracle, MySQL, Linux, and Location by the workflow demo project, and placed into the C:/temp directory for IDM2 Delimited text driver to pick up:

- **Location**:
  ```xml
  <?xml version="1.0" encoding="UTF-8" ?>
  <nds dtdversion="1.1" ndsversion="8.6">
    <input>
      <modify class-name="User" src-dn="cn=user2,ou=Users,o=Novell">
        <association>cn=user2,ou=Users,o=Novell</association>
        <modify-attr attr-name="L">
          <remove-all-values />
          <add-value>
            <value>Toronto</value>
          </add-value>
        </modify-attr>
      </modify>
    </input>
  </nds>
  ```

- **Linux**:
  ```xml
  <?xml version="1.0" encoding="UTF-8" ?>
  <nds dtdversion="1.1" ndsversion="8.6">
    <input>
      <modify class-name="User" src-dn="cn=user2,ou=Users,o=Novell">
        <association>cn=user2,ou=Users,o=Novell</association>
        <modify-attr attr-name="linux">
          <remove-all-values />
          <add-value>
            <value>FALSE</value>
          </add-value>
        </modify-attr>
      </modify>
    </input>
  </nds>
  ```

- **MySQL**
  ```xml
  <?xml version="1.0" encoding="UTF-8" ?>
  <nds dtdversion="1.1" ndsversion="8.6">
    <input>
      <modify class-name="User" src-dn="cn=user2,ou=Users,o=Novell">
        <association>cn=user2,ou=Users,o=Novell</association>
        <modify-attr attr-name="MySQL">
          <remove-all-values />
          <add-value>
            <value>FALSE</value>
          </add-value>
        </modify-attr>
      </modify>
    </input>
  </nds>
  ```

- **Oracle**
  ```xml
  <?xml version="1.0" encoding="UTF-8" ?>
  ```
Importing a driver

To begin, a driver must be “imported” into the system. A “driver” in this sense is a combination of actual program code and objects within the directory that make up the total driver. The import process creates the necessary objects within the directory and provides some preconfigured logic to work with. Usually a particular import will also have the particular code being called and some preconfigured parameters for the driver. The objects created within the directory represent the driver, any associated rules for the driver and the channels (publisher and subscriber) used by the driver. Also all of the configuration information about the driver is stored within the directory on these objects.

For this solution the Delimited Text driver provided with the IDM2 installation is going to be imported. To begin, login to iManager, the page below should appear.
• Under the DirXML Utilities heading is a link called “Import Drivers”. Select that link.
• The Import Driver wizard will appear. It starts off asking if the driver is going to placed into an existing Driverset or a new Driverset. Since this is a new installation a new driverset will need to be created. Select the “in a new Driverset” radio button and then select “Next”.

![Novell iManager](https://example.com/novell-imag.png)
• A new screen in the wizard will appear prompting for the name and context of the new Driverset and server to associate with the driverset. Keep in mind that a server can only be associated with one driverset at a time. The “create a new partition on this drivetset” checkbox can be left checked. This will make the driverset container a partition within the directory. Click “Next” to continue.

• After the Driverset is created, the Import Wizard will bring up the Driver Import page. From here any of the preconfigured drivers or IDM2 policies can be imported into the newly created driverset. For this solution the only driver needed is the Delimited Text - CSV driver. Place a checkmark next to it and click “Next”.
The Driver parameter page will appear. All of the parameters can be changed after a driver is imported but this gives the user a chance to do it initially. The following information needs to be populated on this page:

- Driver Name: Delimited Text
- Output path (this is for the subscriber channel) - c:\temp\delimited\sub
- Output file extension: .xml
- Input path (this is for the publisher channel) - c:\temp\delimited\pub
- Input file extension: .xml
- Enter the name of a container for new users - select any container: users.org. The driver is never going to get an ADD event on the publisher channel so it does not matter in this case.
- Configure data flow: bi-directional. The data needs to flow in both directions.
- Install driver as Remote/Local: LOCAL. The driver is going to be local in this case. A remote loader allows the driver to be placed on a system different than the engine. This is sometimes preferable for some security situations or system load issues.

- The rest of the parameters have to do with using the remote loader which in this case is not needed.
The final part of the import is to define security equivalences and administrative roles.

- Security Equivalence grants the driver the same rights as the selected roles or user objects. For this example, the admin object should be used. However in a production environment an object with more limited rights with the directory may be preferable.

- The Administrative roles lets the user select objects that they want to exclude from synchronization. In this case the admin user object should be excluded from replication.

- After click defining the above roles and clicking “Next” the new driver will be imported into the directory. If the import is successful the following screen will appear:
• Click “Finish” to return the main iManager screen.

Congratulations! The Delimited Text driver is now imported and the configuration of the driver can now begin.

Configuring the Driver
The driver can be configured in any particular order. For this example, the subscriber channel will be configured first followed by the publisher channel.

Subscriber Channel Settings
On the subscriber channel the following logic needs to be implemented:
• Only allow ADD events when the email address and manager attributes are populated.
• Send an email to the newly provisioned person to invite them to sign up for resources
• Create an ADD-ASSOCIATION event to populate the association within the directory. For this solution the users dn will be used for the association value. This must be populated before the user object will be updated.

**Modification of the driver can be accomplished from the DirXML Overview page.** From the main iManager page, expand the DirXML Management heading and select the “Overview” link. A dialog page will appear prompting for the driverset object to look at. Clicking the “Search” button will cause the wizard to search the tree for all driversets. Since there is only one in the tree at this point it will automatically select it and go there.

The overview page outlines the drivers associated with this driverset and what server the driverset is currently associated with. Notice that the Delimited Text driver is represented here. The red circle indicates that the driver is current stopped (Note that the symbol will change depending on the state of the driver. A green circle occurs when the driver is running). Clicking on the red circle will provide a menu for stopping, starting and examining the driver’s properties. The properties option is where
the driver parameters can be modified. These are same parameters that were seen during the driver import. To modify the driver business logic select the Driver’s graphic to the left of the red circle. This will bring up the Driver Overview page (see below).

The driver overview page shows the different policies that are in place for the driver. Hovering over the different objects in the diagram will list their names.

**Limiting ADD events**

To address the first bullet point we need to modify the drivers create policy. Selecting the create policy object will bring up the create policy list dialog box. In this case we want to create a new policy so click on “Insert”. The new policy creation dialog box will appear. Select the following:

- Give the policy a name like: Add event requirements
- The selected container should be the subscriber container already filled in.
- Under “How do you want to implement this policy?” select “Policy Builder”
- Click on “OK” when done
When “Ok” is pressed an empty policy builder window will open.

From the policy builder we can place the logic necessary to block ADD events that do not conform to our specifications. In this case we do not want to allow ADD events that do not have the attributes that we specify: Given Name, Surname, manager, Internet Email Address. Once we are done creating our policy clicking “OK” will save it.
We need to follow a similar procedure for the other requirements on the subscriber channel.

**Sending Email notification**

Email notification will be sent when the ADD event reaches the “Command Transform” policy. To begin create a new policy on the command transform. Then create a new rule. The rule should state in the case of an ADD event send an email. The email command requires the following fields: To, From, Subject, and Message. Also the smtp server will need to be specified as well as any authentication information for that server.
Creating an Add-Association
The DirXML association attribute needs to be populated so that the object thinks it's associated. This will allow the XDS documents that appear on the publisher channel to find their associated eDirectory object. An additional policy on the “Command Transform” will be used to populate the association value. For this example the eDirectory distinguished name (DN) value will be used as the association value. The resultant rule looks like the following:
The following additional modifications should be made to the subscriber channel:

- The filter should be modified to allow through the attributes that are being checked in the create policy.
- A blocking stylesheet should be implemented on the Event Transform to keep the driver from processing events that are not of any concern for the driver. In this case this would be move, delete, rename and sync.
- The schema mapping policy should be cleared. No mapping is needed because the attributes being used are all in the eDirectory namespace.

The filter can be modified by selecting either filter object from the overview and adding and removing the appropriate attributes from synchronization. The filter for this example looks like the image below:
The Blocking stylesheet is implemented on the “Event Transform” policy. Creating a new policy and specifying a XSLT stylesheet instead of the policy builder will bring up a default stylesheet template. A single line is all that’s needed to accomplish the blocking that’s required for this example. The resultant stylesheet is pictured below:
Publisher Channel Settings

On the publisher channel the following logic needs to be implemented:

- Send an email to the person notifying them of results of the workflow.
- Modify the appropriate attributes with the directory. In this case a set of 4 boolean attributes used to signify resource status.

There is not much to be modified on the publisher channel. The only event that will ever occur in this example is a "modify" XDS document that is generated by the workflow. A blocking stylesheet can be implemented in the publisher "Event Transform" as a precaution but it is not explicitly necessary. Since incoming document is an XDS document the modification of attributes will occur if they exist within the directory and on the user (which they will if the schema was modified as outlined earlier). Therefore the only thing that needs to be handled is the email notification to the user with the results of the workflow. The rule developed for this is pictured in the picture above. It is in its own email policy that is part of the "Event Transform".
Configuration of the driver is complete at this point. The driver can be started by going to the overview page, selecting the red circle and clicking on “Start Driver”. If everything is right the driver will start and the circle will change to a green color. Confirm that the logic implemented in the driver is correct by testing the expected inputs and outputs.
PRODUCT DOCUMENTATION

For product documentation, see http://www.novell.com/documentation/

Novell, the Novell logo, NetWare, ConsoleOne, DirXML, GroupWise and NDS are registered trademarks, and eDirectory, exteNd, exteNd Composer, exteNd Director, the N logo and Nsure are trademarks of Novell, Inc. in the United States and other countries.

* Java, Sun and Solaris are registered trademarks and J2EE, JavaMail and JavaServer are trademarks of Sun Microsystems, Inc. AS/400, iSeries, Lotus Notes, OS/390 and zSeries are registered trademarks of International Business Machines Corporation. UNIX is a registered trademark of X/Open, Ltd. Microsoft and Windows are registered trademarks of Microsoft Corporation. MySQL is a registered trademark of MySQL AG. Linux is a registered trademark of Linus Torvalds. Oracle is a registered trademark of Oracle Corporation. All other third-party trademarks are the property of their respective owners.