QAD Enterprise Applications

Administration Guide
System Security

Introduction to System Security and QRA
Security and User Management in QAD Enterprise Edition
Reverse Proxy
Secured Resources Configuration

70-3345-2016EE-Rev1
QAD Enterprise Edition 2016
May 2016

SystemSecurity_AG_v2016EE.pdf/sti/b3s

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
http://www.qad.com

Contents

System Security Change Summary

The following table summarizes changes to this document.

Date/Version	Description	Reference
May 2016/2016 EE-Rev1	Added parameter explanations when installing a DSML Gateway or adding multiple LDAP services.	page 9
	Added Reverse Proxy chapter.	page 17
	Expanded Secured Resources chapter to include Resource Bill of Materials Tool for identifying missing permissions and dependent resources.	page 32
	Moved Glossary terms to the QAD Glossary on the QAD Document Library: documentlibrary.qad.com	
	Multiple minor edits and updates.	
March 2016/2016 EE	Updated with secured resources and YAB details.	
September 2015/2015.1EE	Updated to include Channel Islands UI Permissions and Security Controls for internal use.	
March 2015/2015EE	Initial internal release.	

Chapter 1

Introduction to System Security and QRA

This guide describes the system security features in QAD Enterprise Edition. These features require the QAD Reference Architecture (QRA).

Overview

The QRA-based system security includes security and single sign-on (SSO) features and provides user synchronization and user authentication capabilities.

QAD Reference Architecture

QAD Enterprise Edition now includes a layered, services-oriented architecture called the QAD Reference Architecture (QRA). This architecture works in conjunction with the standard QAD Enterprise Edition architecture to provide additional capabilities. The features described in this guide require QRA.

If you are using the Channel Islands UI with QAD Enterprise Edition, some administration tasks will require familiarity with the concepts of QRA, including modules, business entities, and the framework for running the QAD Enterprise Edition system.

QRA is a layered, services-oriented architecture, with separate layers responsible for presentation, business logic, data, and foundation services:

- Presentation — the presentation layer contains the components that implement the user interface.
- Business — the business layer contains the application business logic and exposes functionality to consumers through published business service APIs.
- Data — the data layer represents the data store of the application.
- Foundation — the foundation layer contains functionality and services that are common to multiple architectural layers (such as exception handling, logging, and so on).

QRA has a modular architecture, and each of the business areas of QAD's products is organized in modules. Modules are comprised of object-oriented business entities.

YAB

QAD Enterprise Edition is installed using the YAB installer and is managed using the YAB console. YAB is a deployment and management toolset that covers all products installed into an Enterprise Edition environment, including the QAD .NET UI.

QAD Applications and Authentication

QRA-based security provides user synchronization and user authentication for the QAD applications listed in Table 1.1.

The table includes the following columns:

- Application — the QAD application.
- Access — indicates whether the user is an internal user or an external user such as a customer.
- Application Type — indicates whether the QAD application is a native application, a web application, or "application to application" (A2A). A native application is launched as an application on Windows, such as the QAD .NET UI. A web application is hosted on a web site

and accessed using a web browser. An A2A communicates directly with another application. For example, the Supply Chain Portal (SCP) Poller communicates directly with QAD Enterprise Edition (QAD base).

- QAD Default Authentication — the default authentication method that is used by the application without QRA-based security.
- QRA Security Authentication — the authentication method used by the application with QRA-based security.

Table 1.1 QAD Applications

Application	Access	Application Type	QAD Default Authentication	QRA Security Authentication
QAD Enterprise Applications Enterprise Edition (QAD base)	Internal	Native	Internal	LDAP
QAD .NET UI	Internal	Native	QAD base / Active Directory	LDAP
Character UI (CHUI)	Internal	Native	QAD base	LDAP
Customer Self Service (CSS)	External Internal	Web	Internal	SAML
Supply Chain Portal (SCP) - Poller Only	Internal	A2A	QAD base	Digital Certificate
QXtend	Internal	A2A / Web	QAD base (A2A) / Tomcat (Administration)	Digital Certificate SAML
Business Intelligence (BI)	Internal	Web	Tomcat	LDAP
Demand Planning	Internal	Native	Internal / LDAP	LDAP
Transportation Management System (TMS)	Internal	Web	Internal	SAML
MQ1 Elements	Internal	Web	Internal / LDAP	LDAP
Business Process Management (BPM)	Internal	Web	QAD base	LDAP
Mobile Field Service (MFS)	Internal	Native	QAD base	LDAP

Note With QRA security, QAD native applications can only be accessed from within the network and do not automatically authenticate based on the Windows credentials.

User Synchronization

User synchronization is the process of synchronizing the user accounts of the multiple QAD applications with the Directory.

User and group synchronization is required to simplify and securely manage information about users on multiple applications. Typically users are centrally managed on an identity management system, and access to applications is enabled through an application management portal. Centralizing the management of user information enables organizations to support the creation, management, and deactivation of users across multiple systems.

The user information in the Directory must be expanded to include information about which QAD applications a user is allowed to access.

Each QAD application must have a unique identifier and the roles must correspond to the roles defined in the applications.

QAD requires a DSML Service as an interface between the various QAD applications and the Directory.

The Directory Services Markup Language (DSML) provides access to Directory Services using SOAP-based web services and represents the directory structure information as an XML document. Version 2 of the standard extends the functionality by providing methods for expressing queries and updates and returns the results as an XML document. DSML services are available from multiple vendors:

- DSML Services for Windows

  http://msdn.microsoft.com/en-us/library/windows/desktop/aa813632(v=vs.85).aspx
- OpenDS

  http://opends.java.net/
- OpenDJ

  http://opendj.forgerock.org/

Communication between the DSML Service and the various QAD applications is based on a SOAP/DSML format. (Communication between the DSML Service and the Directory is based on LDAP).

The information provided to a QAD application from the DSML Service includes a list of all users who can access the application, along with the role memberships of each user.

Each QAD application requires a patch that enables it to communicate to the DSML Service and synchronize user information based on the user information in the Directory.

During user synchronization, a QAD application makes a DSML search request and then receives a DSML search response. Note that any data needed for user provisioning not included in the DMSL response can be supplied by an attribute mapping file.

User Authentication

User authentication is the process by which a user is permitted to log in to a QAD application.

Depending on the QAD application, the authentication process uses the Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language 2.0 (SAML 2.0), or a digital certificate (see Table 1.1, “QAD Applications,” on page 3).

Security Features

The QRA-based security features include the following:

Common Implementation Features

- All network communications are encrypted using SSL.

- Applications support user access management by allowing user accounts to be created, modified, and deactivated.
- Applications support user access management by allowing user accounts to be assigned roles (for example, roles defined in QAD Enterprise Edition).
- Users are uniquely identified by their email address and by their Directory domain and username.
- Applications support auditing by providing username mapping information if the application username differs from the Directory username.
- Each QAD application is assigned a unique group within the Directory.
- Each user ID in the Directory is assigned to the appropriate QAD application group.
- Each QAD application is assigned a collection of roles (for example, roles defined in QAD Enterprise Edition) within the Directory.
- All stored passwords are encrypted or hashed.

Native Application Features

- Native applications use Windows Domain authentication provided by the Directory based on domain, username, and password.
- The SSO user experience does not extend to native applications even when accessed from within the network.
- LDAP connections use SSL (LDAPS).
- LDAP connections are made with a specific service account (username / password).
- LDAP query must be customizable.

Web Application Features

- Web applications use SAML authentication using the identity provider (IdP) for access within the network.
- Applications host a web page with the URL link to the IdP.
- SSO user experience (such as not being prompted to log in again) when accessing other applications protected by the IdP within the same browser session within the network.
- The Service Provider (SP) supports SAML 2.0 profiles and bindings:
  - Profile — urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser
  - Binding — urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
  - Binding — urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect

  Note In SAML terminology, the SP is an external vendor providing the service. In this context, the SP is a QAD application such as TMS or QXtend.
- All SAML 2.0 messages are sent using HTTPS.
- All SAML 2.0 messages are signed with a digital certificate to protect authenticity and integrity.
- Web applications support IdP initiated sign-on.
- Web applications support SP initiated sign-on.
- SP performs a web redirect and not a POST to the IdP.

- SP signs with a 2048-bit certificate.

Chapter 2

Security and User Management in QAD Enterprise Edition

This section covers the following topics:

Overview

This section describes the steps you need to take to configure security and user management in instances of QAD Enterprise Edition managed with the YAB console.

The user management features include the following:

User Synchronization

With user synchronization, QAD Enterprise Edition user accounts can be synchronized with a corporate LDAP directory (Active Directory). The configuration for user synchronization includes the use of a DSML (Directory Services Markup Language) gateway for LDAP communication between QAD Enterprise Edition and a corporate LDAP directory.

LDAP Authentication

Users can log in to QAD Enterprise Edition with their existing corporate credentials. Configuration is done with LDAP Instance Maintenance (36.3.10) and User Maintenance (36.3.1) settings.

E-mail Login

Users can log in with either their user ID or e-mail address and then the same password they use for their other corporate applications.

QRA Session Management

The QAD Reference Architecture (QRA) is a module deployment framework within QAD Enterprise Edition. QRA is required for user synchronization and additional features. QRA session management provides the ability to enable or disable QRA in QAD Enterprise Edition.

As part of this configuration, a secure DSML gateway is required to process all LDAP queries between QAD Enterprise Edition and the corporate Active Directory. Providing a secure connection between the DSML gateway and the corporate Active Directory is essential, so these steps include setting up that aspect of the system's security.

These steps assume that the QAD Enterprise Edition environment has already been installed using the YAB console and that you are familiar with YAB commands. See the *QAD Enterprise Edition Installation Guide* and the *QAD Enterprise Edition Configuration and Administration Guide* for more information.

Server-Side LDAP/Active Directory Authentication

To configure user synchronization in an environment managed with the YAB console:

1. Configure Java Keystore
2. Install DSML gateway
3. Review and Update Directory to QAD Database Mappings

4 Download QAD .NET UI

5 Configure SYNC reason code

6 Configure Alternate Country Code

7 Verify LDAP Instance Definition for DSML Gateway

8 Test User Synchronization

9 Synchronize Users

Configure Java Keystore

A secure connection between the DSML gateway and the Active Directory requires a secure location for signed certificate storage. QAD recommends using a Java keystore (`keystore.jks`) file in a convenient location for the environment. Note the full path to the `keystore.jks` file. You will need it when you configure the DSML gateway.

Follow these high-level steps to configure the Java keystore:

1 Create a keystore file to store the server's private key and SSL certificate.

2 Generate a Certificate Signing Request (CSR) required by the certificate provider.

3 Get an SSL certificate and a Primary Intermediate CA certificate file from a CA vendor (examples include GeoTrust and Verisign/Symantec).

4 Import the Primary Intermediate certificate (primary.p7b) into the Java keystore.

5 Import the SSL certificate into the Java keystore.

6 Validate the keystore entries.

Install DSML gateway

The YAB console supports installing and configuring the DSML Gateway (Configure OpenDJ DSML Gateway).

Goal

Configure one OpenDJ DSML Gateway web app for authentication.

Solution

Add the `opendj-dsml` package to the `build/config/configuration.properties` file and add settings.

1 Add the `opendj dsml` package:

```
packages.opendj-dsml=VERSION
```

2 Configure the basic LDAP settings:

webapp.opendj.ldap.host=. The host name of the underlying directory server.

webapp.opendj.ldap.port=. The LDAP port of the underlying directory server. Default: 636.

webapp.opendj.ldap.usessl=. Indicates whether ldap.port points to a port listening for LDAPS (LDAP/SSL) traffic. true or false

webapp.opendj.ldap.truststore.path=. The trust store used to verify certificates when using secure connections. If you want to connect using LDAPS or StartTLS, and do not want the gateway blindly to trust all certificates, then you must set up a trust store. Not used by default.

webapp.opendj.ldap.truststore.password=. The trust store password. If you set up and configure a trust store, then you need to set this as well. Not used by default.

webapp.opendj.ldap.isactivedirectory=. Designate whether Active Directory is used. true or false

webapp.opendj.ldap.domains=. An optional, comma-delimited list of valid domains.

webapp.opendj.ldap.description=. A description of the OpenDJ Instance.

3 Update your environment. To run only the specific steps related to the DSML Gateway, enter

```
> yab webapp-opendj-update
> yab ldapinstance-opendj-create
```

To update your entire environment, enter:

```
> yab update
```

Setting Up Multiple LDAP Services

The previous settings support a single LDAP service. When multiple are required, use the following steps as an example.

1 Open the `build/config/configuration.properties` file.

2 Add the new opendj instance for a second LDAP service.

```
@extends webapp._base
```

3 Configure the web app settings.

```
webapp.opendj2=
```

Note Do not enter a value for `webapp.opendj2`. This is the YAB configuration syntax for defining a new instance of the webapp type. This example creates “opendj2” webapp. This token can be any valid identifier, but ensure the “type” is defined as “opendj.”

webapp.opendj2.context=. The name of the webapp that gets deployed; in this example, `opendj2`.

webapp.opendj2.application=. A parameter required for YAB. Leave as `${packages.opendj-dsml.dir}`

webapp.opendj2.tomcat=. A parameter required for YAB. Leave as `tomcat.default`

webapp.opendj2.upgrade.includes=. A parameter required for YAB. Leave as `WEB-INF/web.xml`

webapp.opendj2.type=. opendj

webapp.opendj2.ldap.host=. The host name of the underlying directory server.

webapp.opendj2.ldap.port=. The LDAP port of the underlying directory server. Default: 636.

webapp.opendj2.ldap.userdn=. The DN used by the DSML gateway to bind to the underlying directory server.

webapp.opendj2.ldap.userpassword=. The password used by the DSML gateway to bind to the underlying directory server.

webapp.opendj2.ldap.authzidtypeisid=. Required boolean parameter specifying whether the HTTP Authorization header field's Basic credentials in the request hold a plain ID, rather than a DN. This parameter can help you set up the DSML gateway to do HTTP Basic Access Authentication, given the appropriate mapping between the user ID and the user's entry in the directory. If set to true, then the gateway performs an LDAP SASL bind using SASL plain, enabled by default in OpenDJ to look for an exact match between a uid value and the plain ID value from the header. In other words, if the plain ID is bjensen, and that corresponds in the directory server to Babs Jensen's entry with DN uid=bjensen,ou=people,dc=example,dc=com, then the bind happens as Babs Jensen. Note also that you can configure OpenDJ identity mappers for scenarios that use a different attribute than uid, such as the mail attribute.

Default: false

webapp.opendj2.ldap.usessl=. Indicates whether ldap.port points to a port listening for LDAPS (LDAP/SSL) traffic. true or false

webapp.opendj2.ldap.usestarttls=. Leave blank.

webapp.opendj2.ldap.truststore.path=. The trust store used to verify certificates when using secure connections. If you want to connect using LDAPS or StartTLS, and do not want the gateway blindly to trust all certificates, then you must set up a trust store. Not used by default.

webapp.opendj2.ldap.truststore.password=. The trust store password. If you set up and configure a trust store, then you need to set this as well. Not used by default.

webapp.opendj2.ldap.isactivedirectory=. Designate whether Active Directory is used. true or false

webapp.opendj2.ldap.domains=. An optional, comma-delimited list of valid domains.

webapp.opendj2.ldap.description=. A description of the OpenDJ Instance.

4 Update your environment. To run only the specific steps related to multiple LDAP services, enter

```
> yab webapp-opendj2-update
> yab ldapinstance-opendj2-create
```

To update your entire environment, enter

```
> yab update
```

Review and Update Directory to QAD Database Mappings

During user synchronization, QAD Enterprise Edition makes a DSML search request and then receives a DSML search response.

The attributes and values returned in the DSML response typically do not match the fields in the user table associated with the application. For each QAD application, an attribute mapping file, `user-map.xml`, contains a list of attribute-to-field mappings, default values, and processing instructions. This file must be configured to manage the attribute mappings.

Each QAD application performing user synchronization requires a customized attribute mapping file (`user-map.xml`). The file must be customized based on what the particular QAD application requires to provision a user.

Planning is required so that the values in the customized attribute mapping file match what is expected by the QAD application. For example, in QAD Enterprise Edition, active users must have an active user reason code assigned. The reason codes themselves must first be defined using Reason Codes Maintenance (36.2.17).

Mapping between the LDAP directory attributes and the QAD database records is defined in the `user-map.xml` file. Reviewing the `user-map.xml` file requires some familiarity with LDAP attribute usage. You should review the directory using a tool such as Active Directory Explorer or JExplorer. You should also have some familiarity with QAD database tables as described in the *QAD Database Definitions Technical Reference*.

The default `user-map.xml` file is located in a path such as:
`.../build/catalog/packages/mfgcoreplus/n/n/n/n/qad.mfgcoreplus/config/user-map.xml`

If you have trouble locating the file, enter:

```
find . -name "user-map.xml"
```

You must edit this file and define the attribute mappings that are needed based on how the Active Directory is organized.

XML schema for user-map.xml

XML Schema for user-map.xml

Attribute name	Required	Default Value	Description
`name`	true	N/A	The Active Directory attribute name returned in the DSML response.
`tableName`	true	N/A	The table name mapping.
`fieldName`	true	N/A	The field name mapping.
`overwrite`	false	true	If true then the value is overwritten during an UPDATE of the user record. If false then the value is only written during the CREATE of the user record.
`defaultValue`	false	A2A	The default value to use if no value is provided in the DSML response.
`filter`	false	true	Remove attributes that do not match a mapping key. Useful for filtering multiple values such as the attribute `memberOf`.

Map element attributes

Zero or many map elements may be associated with an attribute. If no mapping elements are present, the attribute value will not be replaced.

Map element attributes

Attribute name	Required	Description
key	true	The case insensitive mapping key that is used when matching an attribute value.
value	true	The mapping value that will replace the original value.

Example user-map.xml

```
<user>
    <attributes>
      <attribute name="c" tableName="usr_mstr" fieldName="usr_ctry_code" overwrite="false">
            <map key="U.S.A" value="US" />
            <map key="AU" value="AUS" />
        </attribute>
       <attribute name="mail" tableName="usr_mstr" fieldName="usr_mail_address" overwrite=
"true" />
        <attribute name="c" tableName="usr_mstr" fieldName="usr_lang" defaultValue="US"
overwrite="false">
            <map key="U.S.A" value="US" />
            <map key="AU" value="US" />
            <map key="DE" value="GR" />
        </attribute>
    </attributes>
</user>
```

The above `user-map.xml` file is summarized in the following table.

Active Directory Attribute	Table and Field	Default Value	Is Mapped	Overwrite
c	usr_mstr.usr_ctry_code	N/A	yes	yes
c	usr_mstr.usr_lang	US	yes	no
mail	usr_mstr.usr_mail_address	N/A	no	yes

LDAP Attribute Listing

Attribute Name	Alias	Description	Multiple Values	Syntax
c	countryName	Country abbreviation	false	DirectoryString
cn	commonName	Name	false	DirectoryString
co	friendlyCountryName	Full name of country	false	DirectoryString
codePage	codePage	Code page	false	Integer
countryCode	countryCode	Country code	false	Integer
dn	distinguishedName	X500 distinguished name	false	DN
displayName	displayName	Display Name	false	DN
gn	givenName	First or given name	false	DirectoryString
homePhone	homeTelephoneNumber	Home phone number	false	DirectoryString
mail	rfc822Mailbox	E-mail address	false	DirectoryString
memberOf	memberOf	Group membership	true	DN
mobile	mobileTelephoneNumber	Mobile phone number	false	DirectoryString
modifyTimestamp	mnodifyTimestamp	Modify time stamp	false	Generalized Time
o	organizationName	Organization name	true	DirectoryString
objectCategory		Object category	false	DN
ou	organizationalUnitName	Usually department or sub-entity	true	DNWithBinary

Attribute Name	Alias	Description	Multiple Values	Syntax
postalCode	postalCode	Post code or ZIP	false	DirectoryString
sn	surname	Surname or last name	false	DirectoryString
st	stateOrProvinceName	State	false	DirectoryString
street	streetAddress	Street address	false	DirectoryString
uid	userid	Username	false	DirectoryString

Syntax

Attribute Name	Format	Description	Example
Generalized Time	YYYYMMDDHHMMSS[.|,fraction][(+|-HHMM)|Z]	Time stamp	"19991106210627.3Z" = Nov 6, 1999 21:06:27.3 UTC
DN	cn=<value>,ou=<value>,o=<value>,c=<value>	Distinguished name. Comma delimited list of name/value pairs (RFC 2253)	cn=Ben Gray,ou=editing,o=New York Times,c=US
DirectoryString		UTF-8 encoded string	QAD Inc.
Integer		Whole number of unlimited magnitude	12345

Once the attribute mapping in `user-map.xml` is complete, you can proceed with configuring QAD Enterprise Edition.

Download QAD .NET UI

Download and open the QAD .NET UI client from the home server URL.

Verify that the system is working. Open a program (such as Sales Order Maintenance [7.1.1]), a browse (such as Browse Master Browse [36.4.8.14]), and a Financials function (such as Role Membership Maintain [36.3.6.6]).

Configure SYNC reason code

Open Reason Codes Maintenance (36.2.17) and set the following fields:

1. Reason Type: USER_ACT
2. Reason Code: SYNC
3. Click Next.

Configure Alternate Country Code

Depending on the data you have, Alternate Country Code might not be set, which can cause problems when you try to add a user. For each country where you have users, add an Alternate Country Code using Country Code Data Maintenance (2.14.1).

For example, for a US user, open Country Code Data Maintenance, set Country Code and Alternate Country Code to US, click Active on, and then click Next.

Verify LDAP Instance Definition for DSML Gateway

The YAB console configures the DSML gateway instance for you in QAD Enterprise Edition. You can verify the configuration as follows:

1 In the QAD .NET UI, open LDAP Instance Maintenance (36.3.10) and check the following fields:

- LDAP Instance Name: `opendj`
- Description: OpenDJ Instance – (or some appropriate description)
- LDAP Servlet URL: `http://domain:port/opendj/DSMLServlet` (where domain is the Tomcat server)
- LDAP Domains: leave blank.
- Is Active Directory: check on.

2 Click Next.

Test User Synchronization

Next, test user synchronization by adding a user. The following example adds the user `abc` (and then also uses the `abc` user's credentials to synchronize against the directory). To add another user, such as `def`, just enter (`cn=def`) in the Update Search Field below.

The Update Search Filter field identifies the user being added (`abc`), while the Search UserID field identifies a user that can authenticate against the directory to get access to the directory data. In the following example, the user `xyz` is used to authenticate against the directory.

Note that the specifics for the Update Search Root and SearchUserID settings depend on how your LDAP directory structure is configured.

1 Open Active Directory User Sync (36.3.11), and set the following fields.

Note Use your user ID rather than abc, and your password.

- Create Users: `on`
- Update Users: `on`
- Update Search Filter: `(cn=abc)`
- Update Search Root: `OU=Users,OU=Accounts,DC=qad,DC=com`
- Update Search by Group: `off`
- Deactivate Users: off (and subsequent Deactivate-related fields are blank)
- LDAP Instance Name: (as specified in LDAP Instance Maintenance)
- Search UserID: `CN=xyz,OU=Users,OU=Accounts,DC=qad,DC=com`
- Search User Password: (enter the password for the user specified in Search UserID)

2 Click Next.

Configure User

Verify that the user has been added by opening User Maintenance (36.3.1) and looking up the user. The fields should be populated correctly; for example, User Name should show the full name, Remarks should include information about the user's job, and so on.

Open User Domain/Entity Access Maintain (36.3.4). Set domain and default domain for your user (for example, `abc`).

Open Role Membership Maintain (36.3.6.6) and specify the roles for your user.

Log in to QAD .NET UI as an LDAP User

Now you can securely log in to the QAD .NET UI using your QAD user ID and password.

Synchronize Users

Now you can use Active Directory User Sync (36.3.11) to create, update, or deactivate users as specified by the search filter settings.

Activity is logged in the `ldapsync-users.xml` and `ldapsync-exceptions.xml` log files.

Chapter 3

Reverse Proxy

This section covers the following topics:

Overview

Exposing any application to the internet greatly increases the security risks faced by that application. To provide secure access to external web applications, you can use a QAD-supplied generic proxy that hides external applications behind a firewall, ensuring that the secure Channel Islands UI controls access to those applications.

Configuration

The Proxy Controller is configured in `qad-webshell.properties`.

Proxy Configuration

Each external application has a name, as well as a local path from which to proxy, and a remote server to which to proxy:

Table 3.1
Proxy Configuration

Property	Description
qad-webshell.proxy.names	Comma-separated list of proxy names. These are then used as keys for the other properties in this table.
qad-webshell.proxy.{name}.proxyFrom	The path from which to proxy; for example, /remote must start with a /. The full proxy path is generated from this value as /{context}/proxy{proxyFrom}.
qad-webshell.proxy.{name}.proxyTo	The URL to which to proxy; for example, https://remote.qad.com/context.

Note If the external application is hosted at the root of the remote server, you must include a forward slash, `/`, at the end of the `proxyFrom` and `proxyTo` parameters. For example, to proxy to the Tomcat default application https://vmj8e01.qad.com:22000, use:

```
proxyFrom=/tomcat/ and proxyTo= https://vmj8e01.qad.com:22000/
```

Without this forward slash, relative links on the page may not be resolved correctly.

URL Rewriting

With a reverse proxy, it is often necessary to rewrite URLs in the proxied content to be consistent with the local path. For example, the local path /qad-webshell/proxy/app is proxied to the remote server https://remote.qad.com/context, and the proxied content contains links similar to the following

```
<a href="https://remote.qad.com/context/some/cool/api">Click Me!</a>
<a href="/context/some/cool/api">Click Me!</a>
```

Without rewriting, the first link would try to bypass the proxy and go directly to the remote server. The second link would resolve as:

```
https://webshell.qad.com/context/some/cool/api
```

when it should instead be:

```
https://webshell.qad.com/proxy/app/some/cool/api
```

The solution to these problems is URL rewriting. Currently, QAD supports rewriting for HTML (html), JavaScript (js), and JSON (json) content. To enable content rewriting with default settings, add the following property with a comma-separated list of rewriters:

qad-webshell.proxy.{name}.rewriters

For example,

```
qad-webshell.proxy.{name}.rewriters=html,json
```

HTML Rewriting

Links in HTML can occur in one of the following places:

- Inside an HTML attribute: `<a href="URL">`
- Inside an HTML event handler: `<body onload="load('URL')">`
- Inside styles attribute or content: `<div style="background-image: url('URL')">` or `<style> { background-image: url('URL'); }</style>`
- Inside script content: `<script>var url = 'URL'</script>`

Enabling the HTML rewriter enables only attribute rewriting by default. To enable and configure link, event, style, and script rules, add the following property with a comma-separated list of rewrite options:

qad-webshell.proxy.{name}.rewriters.html

Each option works by applying a set of rewrite mappings to selected content in the HTML document. Each option has the following default behavior.

link

The link option is responsible for rewriting links in HTML attributes. By default, it looks for links in href, src, action, data, and cite attributes. It applies a set of rewrite mappings to those attribute values in the document. Rewrite mappings are either simple mappings that match and replace a given prefix, or regex mappings that perform general match and replace. The default mappings are based on the proxy configuration. For example, suppose proxyFrom=/app, proxyTo= https://remote.qad.com/path, and the webshell context is qad-webshell. Two simple mappings are defined:

```
/path -> /{context}/proxy/app
https://remote.qad.com/path -> /qad-webshell/proxy/app
```

event

The event option is responsible for rewriting links in HTML event handlers. By default it looks for links in the following attributes:

- onclick
- ondblclick
- onmousedown

- onmouseup
- onmouseover
- onmousemove
- onmouseout
- onkeypress
- onkeydown
- onkeyup
- onfocus
- onblur
- onload
- onunload
- onsubmit
- onreset
- onselect
- onchange

Because the event handler can contain arbitrary JavaScript, this option uses regex mappings by default to look for links in JavaScript strings. Using the same scenario as in the link example, suppose proxyFrom=/app, proxyTo=https://remote.qad.com/path, and the webshell context is qad-webshell. The default mappings are defined:

```
'(?:https://remote\.qad\.com/path|/path)((?:[^'\\]|\\.)*)' -> '/qad-webshell/proxy/app$1'
"(?:https://remote\.qad\.com/path|/path)((?:[^"\\]|\\.)*)" -> "/qad-webshell/proxy/app$1"
```

style

This option rewrites links in script attributes and script content. By default, it uses the following regex mappings to match CSS URL data types:

```
url\(\s*'(?:https://remote\.qad\.com/path|/path)((?:[^'\\]|\\.)*)'\s*\) -> url('/qad-
webshell/proxy/app$1')
url\(\s*"(?:https://remote\.qad\.com/path|/path)((?:[^"\\]|\\.)*)"\s*\) -> url("/qad-
webshell/proxy/app$1")
```

script

This option rewrites links in script content. By default, it uses the same regular expressions as the event option.

The default behavior for each of these options can be customized using the following properties.

Table 3.2
Script Content Properties

Property	Description
qad-webshell.proxy.{name}.rewriters.html.{option}.attributes	Comma-separated list of attributes for link and event options. Include ‘default’ in the list to use default attributes in addition to custom ones.
qad-webshell.proxy.{name}.rewriters.html.{option}.mappings	Comma-separated list of mapping types to apply. Current supported values are ‘simple,’ ‘regex,’ and ‘default’

Property	Description
qad-webshell.proxy.{name}.rewriters.html.{option}.mappings.simple	Simple rewrite mappings in pairs separated by spaces. For example, /path /qad-webshell/proxy/app https://remote.qad.com/path /qad-webshell/proxy/app
qad-webshell.proxy.{name}.rewriters.html.{option}.mappings.regex	Regex mappings in pairs separated by spaces. For example, '(?:https://remote\.qad\.com/path|/path)((?:[^"\\\\]|\\\\.)*)' '/qad-webshell/proxy/app$1' \"(?:https://remote\.qad\.com/path|/path)((?:[^"\\\\]|\\\\.)*)\" "/qad-webshell/proxy/app$1"

Example

In this example, two proxy endpoints are set up:

- /qad-webshell/proxy/configurator -> https://vmwgj03.qad.com:16560/procon
- /qad-webshell/proxy/tomcat/ -> https://vmj8e02.qad.com:22000/

The Tomcat proxy has URL rewriting enabled for HTML links with custom attributes and mappings. The qad-webshell.properties file would look like the following:

```
qad-webshell.proxy.names=tomcat,configurator
qad-webshell.proxy.configurator.proxyFrom=/configurator
qad-webshell.proxy.configurator.proxyTo=https://vmwgj03.qad.com:16560/procon
qad-webshell.proxy.tomcat.proxyFrom=/tomcat/
qad-webshell.proxy.tomcat.proxyTo=https://vmj8e01.qad.com:22000/
qad-webshell.proxy.tomcat.rewriters=html
qad-webshell.proxy.tomcat.rewriters.html=link
qad-webshell.proxy.tomcat.rewriters.html.link.attributes=
href,src,action,value
qad-webshell.proxy.tomcat.rewriters.html.link.mappings=simple
qad-webshell.proxy.tomcat.rewriters.html.link.mappings.simple=/ /qad-
webshell/proxy/tomcat/
```

Chapter 4

Secured Resources Configuration

This section applies only if you are using QAD Enterprise Edition with the Channel Islands UI. It covers the following topics:

Overview

Creating and maintaining a secure environment for the Channel Islands UI requires additional steps in the QAD .NET UI. Secured Resources are QRA modules that provide that security through their associated browses, business entities, services, and views.

Prerequisites

Your system must have a YAB environment set up and running. You must be using the Channel Islands UI.

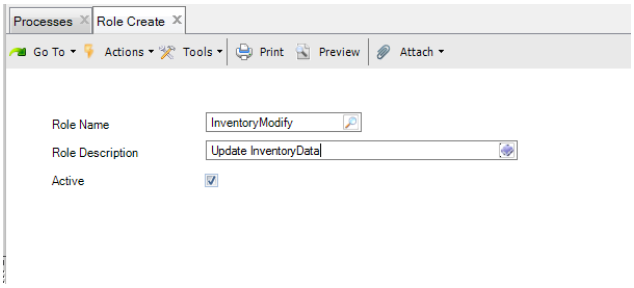
Roles

Roles can be created in QAD Enterprise Edition and assigned to users, providing the proper access needed for users to complete a particular job function. Permissions are configured for each role, setting access to the menus, browses, lookup tables, and dashboard panels that users see, create, edit, and delete, both in the QAD .NET UI and in the Channel Islands UI.

Creating Channel Islands UI Roles

To create a role for the Channel Islands UI, open Role Create in the QAD .NET QAD UI.

Fig. 4.1
Role Create



Give the role a name, add a role description, mark the role as active, and save. Once created, the role becomes available in the application to configure security access and permission levels for application resources, such as screens, standard browses, and lookup tables.

Default Roles

Your system arrived with six predefined, pre-configured roles.

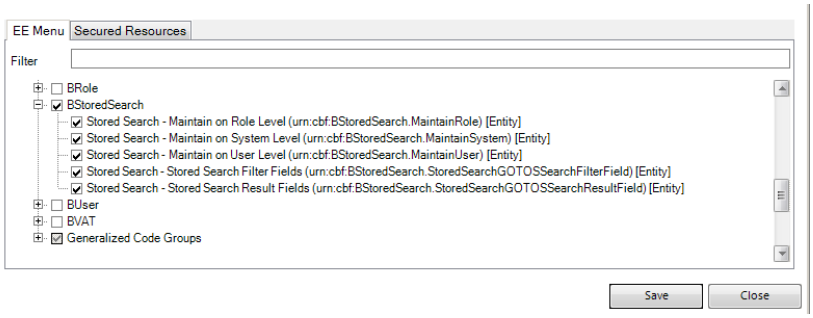
- Customer Service Manager
- Customer Service Representative
- Purchasing Manager
- Sales Manager
- VP of Sales
- webui_user

Configuring Stored Views Access

One feature in the Channel Islands UI is the ability to create and use stored views that customize screen layout and preset search conditions, emphasizing important information needed for everyday use and specific tasks. Depending on the role, you may need to configure different levels of access for the types of stored views a user can create, edit, or delete.

Stored-view security is configured for each role and is completed in Role Permissions Maintain. To enable this feature, double-click a role, then set permissions on the EE Menu tab. Under Secured items not on menu, expand BStoredSearch.

Fig. 4.2
Stored Views



Depending on the access required for this feature, you can grant access to any of the following options, then click Save:

- Stored Search – Maintain on User Level. Allows users to create personal views that only they can view and manage. It is recommended that all users have the ability to customize their own views.
- Stored Search – Maintain on System Level. Allows users to create and manage System views that every user who has access to the Channel Islands UI can view.
- Stored Search – Maintain on Role Level. Allows users to create and manage role views that every user assigned to that role can view.

Assigning Permissions to Roles

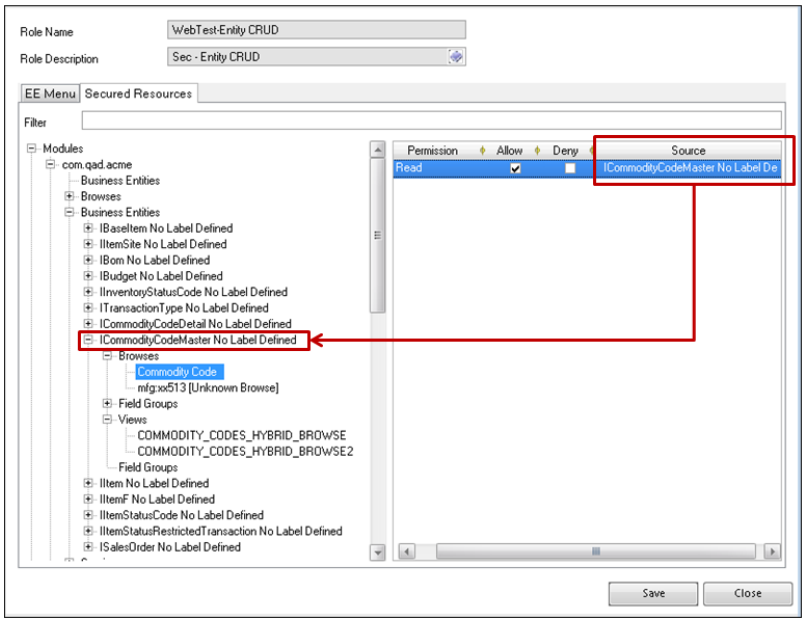
After you create a role, you must configure permissions for the role to allow access to browses, lookup tables, and dashboard panels. All permissions for the Channel Islands UI must be set up and configured using the QAD .NET UI. This process is done on the Secured Resources tab of the Role Permissions Maintain (36.3.6.5) browse. To access the Secured Resources configuration tab, double-click on a role for configuration or review.

Permissions Inheritance and Configuration

Permissions are assigned in a top-down manner, with automatic inheritance for child resources. When you grant Allow access to a parent resource, all children of that resource are granted full permission. You can manage the granularity of child permissions, such as specific lookup tables, by setting those permissions to Deny. If, however, you deny access to a parent resource, all children of that resource are denied access, regardless of what their individual settings are. You cannot set the permission of a child resource to Allow if any permission further up the menu tree has been set to Deny.

When a particular resource has inherited permissions from a parent, the relationship is indicated in the Source column of the Permissions Configuration table on the Secured Resources tab. In Figure 4.3, you see that permissions for the Commodity Code browse resource are inherited from the ICommodityCodeMaster Business Entity resource.

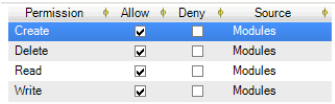
Fig. 4.3
Permissions Configuration Table



Permissions are Create, Read, Write, and Delete. These security settings are fully functional.

Note In order for a user to be able to create, write, and delete, that user also must have read access to the resource.

Fig. 4.4
Available Permissions

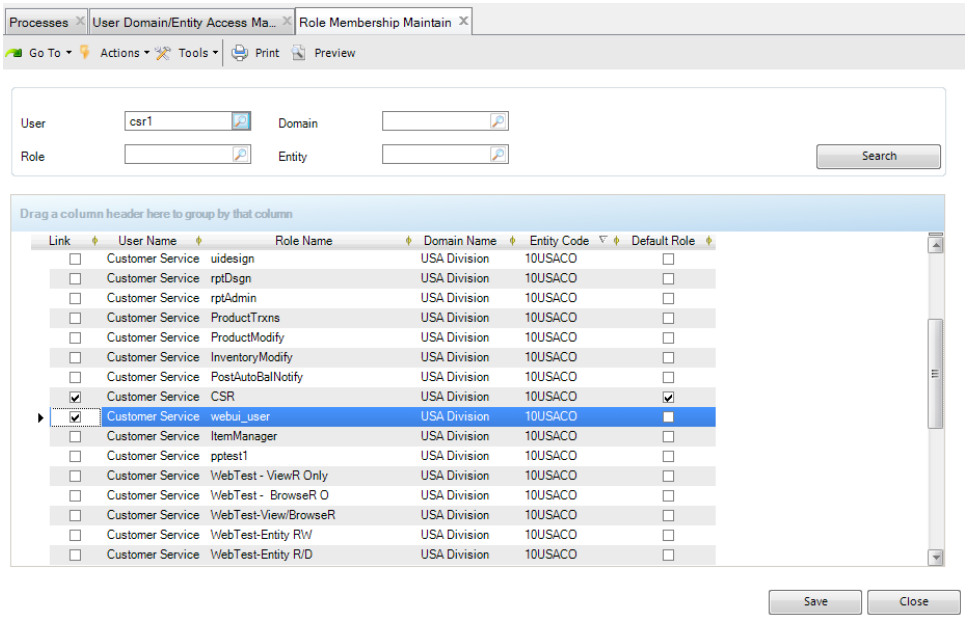


Note For this release, the Channel Islands UI screen design does not change when permissions are denied for a resource, which can lead to user frustration. For example, a role has been denied create access for sales orders. The Channel Islands UI is not going to allow users to actually create and save a sales order, but the New button is still active on the browse and displays a blank sales order screen if users click it. Users can enter all pertinent information, but when they try to save the completed sales order, they get an error that access is denied and they have no way to save the work they have completed. Future releases will address this functionality and align permissions settings to Channel Islands UI design.

Granting Channel Islands UI Access to Users

Every Channel Islands UI user must have at least two assigned roles, one of which is webui_user. A user cannot access the Channel Islands UI without the webui_user role, but can still access the QAD .NET UI. The process of creating a user and assigning roles to that user is identical to the standard procedure described in the *QAD Security and Controls User Guide*, with the additional step of linking the user to the webui_user role on Role Membership Maintain (36.3.6.6).

Fig. 4.5
Webui_user Role



Secured Resources

In the Secured Resources tab of Role Permissions Maintain, you control how application resources are accessed, either from APIs or on the Channel Islands UI. Only resources that are loaded into the application and listed on the Secured Resources tab can be configured as part of role permissions and accessed on the Channel Islands UI. Once resources have been loaded into the application as secured resources, they can be configured for each Channel Islands UI role.

Every Channel Islands UI screen corresponds to a view resource that is identified with a resource URI. The view resource is secured based on the business entity associated with the view. The view may require data from other business entities, such as a master screen's subordinate detail screens, lookups, business services, and APIs. If a user needs access to a given Channel Islands UI screen, that user's role needs access to a variety of other resources for the user to have the full functionality of that screen. If the user's role does not have sufficient permissions for the different resources, the user is presented with an "Error 403: Access Denied, You do not have permission to access the requested page." For information on identifying dependent resources and missing permissions, see "Identifying Dependent Resources and Missing Permissions" on page 32.

Securing Browses, Lookups, and Dashboard Panels

Channel Islands UI browses, lookup tables, and dashboard panels all have corresponding resources listed in the Secured Resources tab in the QAD .NET UI. In order to access and view any of these components in the Channel Islands UI, a role must have the Allow check box selected for the corresponding resources in the Permissions Configuration table.

The distinct Channel Islands UI elements are:

- “Hybrid Browse Screens” on page 28
- “Single Row Edit Grids” on page 29
- “Standard Browse Screens” on page 30
- “Lookups and Dashboard Panels” on page 30

The Secured Resources tab contains the following elements:

- Filter option – allows you to narrow the display of available resources
- Modules

  Within each module are the secured resources building blocks.

  - Browses – The access control to the data behind the associated screen, lookup table, or dashboard panel to which you want to provide access.
  - Business Entities – The business logic and data necessary to represent real world elements, such as sales orders, within the Channel Islands UI.
  - Services – The methods for each business entity as interface services, identified with the entity name and an “i” prefix. Services can be configured so that a role allows or denies access to a method that is associated with the action. Currently the approach is to associate the methods as follows:

Table 4.1 Interface Services

Method	Permission Action
Create	create
Update	write
Delete	delete
Fetch	read

Note Any other methods are associated with the read action.

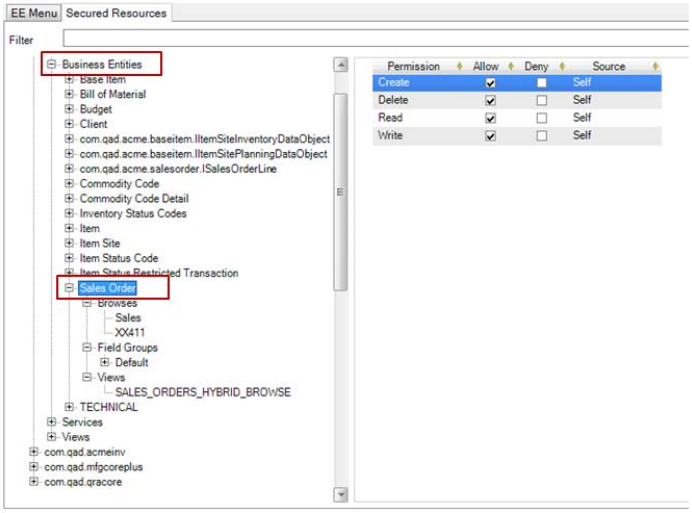
- Views – The access control for the Channel Islands UI drop-down menus that users see and with which they interact. When view resources for hybrid browses and standard browses have Read permission, the associated menu items, such as Sales Orders, become available on the menu bar in the Channel Islands UI.

Hybrid Browse Screens

Hybrid browse screens allow you to view both a static data table and the table’s associated, interactive elements, such as a requisition and that requisition’s lines. The secured resources for hybrid browses are located in the Business Entities branch of a module’s secured resources tree, as illustrated in Figure 4.6. The associated Browse Resources, View Resources, and Field Groups are

collected and can be secured from here. Permissions for hybrid browse screens must be set and managed from the business entity level in order to provide, at a minimum, read access to the screen.

Fig. 4.6
Hybrid Browse



Note This level of permissions is only for the master hybrid browse screen entity. Associated lookup tables that have not been linked to the business entity in the BrowseResource XML file and business entities for related single row edit grids must be configured separately.

Note For the initial Channel Islands UI release, hybrid browses should have full create, write, read, and delete (CRWD) access.

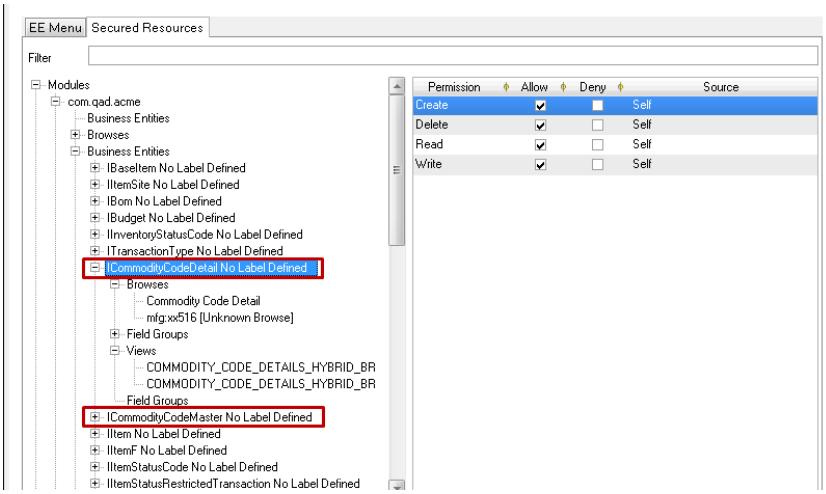
Single Row Edit Grids

Once permissions have been set for hybrid browse screens at the business entity level, some of the screen elements may require additional permissions configuration to ensure complete access to all screen elements on the hybrid browse screen. This includes single row edit grid business entities.

Single row edit grids have their own hybrid browse screens accessible in the grid details window, which means they have their own Business Entity resources that require permission configuration. These resources must be identified and configured in addition to the master entity.

In Figure 4.7, the ICommodityCodeDetail business entity is the entity for the Commodity Code Items Grid on the ACME Commodity Code Screen. Permissions must be set for the Master entity and the Details entity to provide complete access to the screen.

Fig. 4.7
Single Row Edit Grid



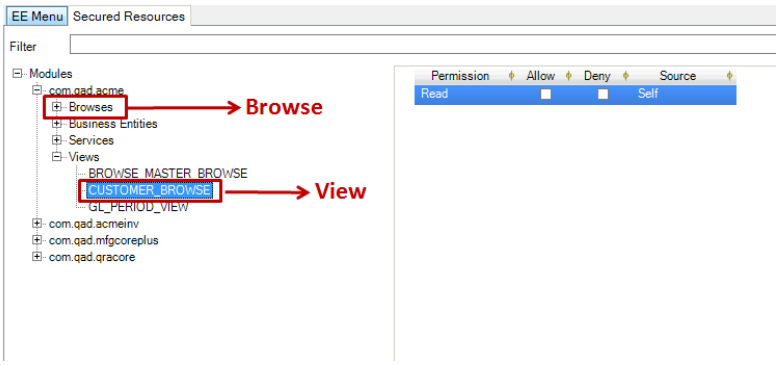
Standard Browse Screens

Standard browses are the QAD .NET UI, Progress-based ".p" browse programs that serve as power browses or lookup browses. A standard browse displays data in a read-only browse table. You cannot edit or delete the existing data, nor add additional records to the browse table. You can filter the view.

Before a user can open a standard browse and view its data, you must assign the correct read permissions to both the associated Browse and View resources. Permission to the Browse Resource gives the user access to the data that loads into the screen, and permission to the View Resource makes the screen available in the Channel Islands UI menus.

To configure access, you must find the associated Browse Resource in the Browses branch of the module tree in the Secured Resources tab and the associated View Resource in the Views branch of the module tree in the Secured Resources tab.

Fig. 4.8
Standard Browse Screen



Lookups and Dashboard Panels

Data linked to lookup tables and dashboard panels are also secured resources. In some cases, these resources are used in multiple places across the Channel Islands UI, such as with a Business Entity and a lookup table on a different screen. When configuring permissions for these resources, access

may already have been granted from another place. However, if a lookup table is not associated with a Business Entity, you must set its permissions to read access or users receive an access denied message when they attempt to open the lookup table from the screen. When a dashboard panel does not have read access, the system displays “NO_DATA_RETURNED” in the panels.

Note This message does not always indicate that access is not configured for dashboard panels. The message also displays if there is actually no data in the back end.

Note Lookup tables and dashboard panels only require permissions configuration in the associated Browses section of the Secured Resources tab. These screen elements do not have associated view resources.

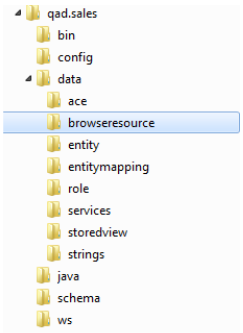
Manually Adding Secured Resources

For detailed information on YAB installation and administration, see the *QAD Enterprise Edition Installation Guide* and the *QAD Enterprise Edition Configuration and Administration Guide*.

To manually add secured resources to enable them to be visible in the Channel Islands UI, follow these steps.

1. Go to the current version subdirectory of the package and into the module/data/browseresource subdirectory. In Figure 4.9, the module is qad.sales.

Fig. 4.9
Browse Resource Subdirectory



2. Open the browseresources.xml file.
3. Add the browse resource to the .xml file. The basic format of a resource is:

```
<BrowseResource>
   <BrowseURI>urn:browse:mfg:xx410</BrowseURI>
   <BusinessEntityURI>urn.be:com.qad.acme.salesorder.ISalesOrder</Bus
   inessEntityURI>
</BrowseResource>
```

   Note If you do not have a business entity URI, leave that entry blank. For example, `<BusinessEntityURI></BusinessEntityURI>.`
4. Load the file with the new browse resource with the following YAB command:

```
yab -clean metadata-<package>-update
```

where `-clean` forces a reload and <`package`> is the actual package name. For example:

```
yab -clean metadata-base-update
```

The browse should now appear in the Secured Resources tab in the QAD .NET UI's Role Permissions Maintenance program.

Identifying Dependent Resources and Missing Permissions

This section documents methods that you can use to identify resource dependencies and missing permissions for browses, single row edit grids, lookup tables, drilldowns, and dashboard panels.

Using the YAB Resource Bill of Materials Tool

The following set of YAB processes can help you identify resource dependencies for view and business entity resources.

Installation

The Resource Bill of Materials (BOM) tool is automatically installed in most YAB environments. To determine if it is present in your system, enter:

```
> yab config packages.resource-bom
```

The Resource BOM is already installed if you see:

```
packages.resource-bom=VERSION
BUILD SUCCESSFUL (1.939 s)
```

If you do not see the build successful message, you can find the Resource BOM tool in the QAD Package Repository under resource-bom. The tool is installed like other YAB modules. To install the module in an existing YAB instance:

1. Add the resource-bom package to the YAB configuration in build/config/configuration.properties:

   ```
   packages.resource-bom=VERSION
   ```

2. Install the package by updating your environment. To run only the steps related to the Resource BOM Tool, enter:

   ```
   > curl -o resource-bom-VERSION.zip http://packages.qad.com/packages/resource-bom/VERSION/
   > yab -i:resource-bom-VERSION.zip
   > yab -r
   ```

   To update your entire environment, enter:

   ```
   > yab update
   ```

   This installs the sec-resource-uri and sec-resource-bom tools, which are described in "sec-resource-uri" on page 33 and "sec-resource-bom" on page 33.

Usage

Note You must be a system administrator with full permissions to run the following sequence of commands.

Before you begin, you need:

- The role of the user who has been denied access.
- The menu label that is returning the error.
- The QRA AppServer running.

  ```
  yab appserver-qra-start
  ```

- The user name and password of a QRA administrator with full permissions. You can configure them in `build/config/configuration.properties`:

  ```
  user=<USERNAME>
  ```

  ```
  password=<PASSWORD>
  ```

  The user name and password can also be configured from the command line when running the processes:

  ```
  > yab sec-resource-uri -user:[USERNAME] -password:[PASS] ...
  ```

  ```
  > yab sec-resource-bom -user:[USERNAME] -password:[PASS] ...
  ```

sec-resource-uri

The sec-resource-uri process helps you find resource URIs based on resource label and type. To use this tool, run:

```
> yab sec-resource-uri -label:<Label> -type:<ResourceType>
```

where Label is the name of the menu label that is denying access and ResourceType, if known, is the specific resource of that menu label. See “Identifying Resources by URI” on page 43 for more information on resource types. This process returns the resources in the system, along with their URIs, filtered by input. If you specify the label option, the resources are filtered by label using a case-insensitive contains relationship. Enclose menu labels of two words with quotation marks, retaining the original spacing; for example, -label:“Sales Order.” If you specify type, the resources are filtered based on resource type. If neither label nor type is specified, then all resources defined in the system are returned; for example,

$ yab sec-resource-uri -label:Item -type:view

```
Label                          URI
------------------------------ ------------------------------------------------------------
Customer Item                  urn:view:browse:com.qad.erp.sales.customerItemBrowse
ITEM                           urn:view:hybridbrowse:com.qad.erp.base.items
ITEM_FISCAL_CLASS_MAINTENANCE  urn:view:hybridbrowse:com.qad.erp.base.fiscalClasses
ITEM_SITE_COST                 urn:view:hybridbrowse:com.qad.erp.base.itemSiteCosts
ITEM_SITE_INVENTORY            urn:view:hybridbrowse:com.qad.erp.base.itemSiteInventories
ITEM_STATUS                    urn:view:hybridbrowse:com.qad.erp.base.itemStatuses
Inventory Detail by Item      urn:view:browse:com.qad.erp.sales.inventoryDetailByItemBrowse
Item Replacement               urn:view:hybridbrowse:com.qad.erp.sales.itemReplacements
Item Replacement/Up/Cross-Sell
urn:view:browse:com.qad.erp.sales.itemReplacementUpCrossSellBrowse
Sales by Customer/Item         urn:view:browse:com.qad.erp.sales.salesByCustomerItemBrowse
Sales by Item                  urn:view:browse:com.qad.erp.sales.salesByItemBrowse
Supplier Item                  urn:view:browse:com.qad.erp.requisition.supplierItemBrowse
mfg-ITEM_SITE_PLANNING         urn:view:hybridbrowse:com.qad.erp.base.itemSitePlannings
```

sec-resource-bom

The sec-resource-bom process can help you determine dependent resources and missing permissions. To use this tool, run:

```
> yab sec-resource-bom <URI> <ROLE>
```

where URI is the resource you identified using the sec-resource-uri process, and ROLE is the name of the role that has been denied access.

The output is a view of the access control tree for the given resource and role, with additional nodes added for resource dependencies.

For example:

```
$ yab sec-resource-bom urn:view:hybridbrowse:com.qad.erp.base.items SuperUser
L Label                        URI                                          Perm
- ------------------------- ------------------------------------------- ----
0 ITEM                      urn:view:hybridbrowse:com.qad.erp.base.items R
1 Dependent Entities
2 IItemSite [no label defined] urn:be:com.qad.base.item.IItemSite        CDRW
2 Item                       urn:be:com.qad.base.item.IItem              CDRW
1 Browses
2 Item Master                 urn:browse:mfg:gp340                       R
1 Browse Lookups
2 Code Master                 urn:browse:mfg:gp072                       R
2 Product Line Master         urn:browse:mfg:gp343                       R
2 Item Status Codes           urn:browse:mfg:gp198                       R
2 Item Descriptions           urn:browse:mfg:gp197                       R
2 Item Master                 urn:browse:mfg:gp340                       R
1 Drilldowns
2 Product Line Master         urn:browse:mfg:gp343                       R
2 Item Master                 urn:browse:mfg:gp340                       R
2 Item Status Codes           urn:browse:mfg:gp198                       R
2 Item Descriptions           urn:browse:mfg:gp197                       R
1 Maintenance Lookups
2 Warehouse Item Type         urn:browse:mfg:wh047                       R
2 Item Descriptions           urn:browse:mfg:gp197                       R
2 Code Master                 urn:browse:mfg:gp072                       R
2 Freight Classes             urn:browse:mfg:so001                       R
2 Item Types                  urn:browse:mfg:gp134                       R
2 Replenishment Type          urn:browse:mfg:wh210                       R
2 Item Buyer/Planners         urn:browse:mfg:gp001                       R
2 Secondary Run Sequence      urn:browse:mfg:gp781                       R
2 Routing Detail              urn:browse:mfg:gp220                       R
2 Groups                      urn:browse:mfg:pc001                       R
2 Tax Class                   urn:browse:mfg:tx008                       R
2 Supplier                  urn:browse:fin:BCreditor.SelectCreditor      R
2 Inventory Locations         urn:browse:mfg:gp336                       R
2 Product Line Master         urn:browse:mfg:gp343                       R
2 Issue Method                urn:browse:mfg:wh211                       R
2 Commodity Codes             urn:browse:mfg:tx003                       R
2 Network Codes               urn:browse:mfg:dn002                       R
2 Cost Elements by Element     urn:browse:mfg:gp115                      R
```

```
2 Primary Run Sequence        urn:browse:mfg:gp780                R
2 Unit of Measure Group       urn:browse:mfg:wh067                R
2 Location Types              urn:browse:mfg:gp784                R
2 Configuration Types         urn:browse:mfg:gp325                R
2 Bills of Material           urn:browse:mfg:gp136                R
2 Language Code               urn:browse:mfg:gp900                R
2 Fiscal Class                urn:browse:mfg:gp147                R
2 Item Status Codes           urn:browse:mfg:gp198                R
2 Sites                      urn:browse:mfg:gp348                 R
2 Auto Lot Master             urn:browse:mfg:gp019                R
2 Inventory Status Codes       urn:browse:mfg:gp349               R
```

BOM Components

Level

The level of the resource within the access control tree. By default, only two levels of the tree are displayed; however, you can configure this by adding the -depth:N option to the command.

Label

The label for the resource. The generic container labels are:

- Browses – Stand-alone browses. The BOM may return two Browses listings. The Browse with no URI indicates that the following line is a dependent browse resource that requires permissions be secured. The Browse with a URI is a system-defined container of browses that itself must be secured.
- Field Group – A set of fields that can be secured as a single entity.
- Views – Hybrid browses, which show both a browse and a maintenance screen in a composite view.
- Dependent Entities – The resources required for the screen to function properly. Dependent entities must be secured independently from the main view. The label items directly under Dependent Entities are business entities that have been or need to be secured.
- Browse Lookups – Browses used for lookups in the browse view of a hybrid-browse screen. For the lookup to function correctly, the user must have read permission to the browse.
- Browse Drilldowns – Browses used as drilldowns in the browse view of a hybrid-browse screen. For the drilldown to function correctly, the user must have read permission to the associated browse
- Maintenance Lookups – Browses used for lookups in the maintenance view of a hybrid-browse screen. For the lookup to function correctly, the user must have read permission to the browse.

Some resources may be labeled as “MISSING RESOURCE.” This indicates that a resource is referenced by the input resource but is not defined in the system.

URI

The URI associated with each label item. For more information and definitions of the components of a URI, see “Identifying Resources by URI” on page 43.

Perm

The permissions that are currently set for the associated resource: C (create), D (delete), R (read), W (write). If this column is blank, the resource has not been secured and cannot be accessed in the Channel Islands UI. Use this column to determine which resources are missing permissions and need to be secured.

Using the Results of the BOM Tool

1. Identify which resources have not been properly secured and are missing permissions. In particular, look for resources under Dependent Entities, Lookups, Drilldowns, and Browses that do not have adequate permissions in the Perm column.
2. Log into the QAD .NET UI as a super user and go to Role Permissions Maintain.
3. Double-click the role that needs access.
4. In the Filter field on the Secured Resources tab, enter the label description from the BOM.
5. The filtered result should be the resource requiring permissions. In the Permissions table on the right, select the correct Allow check boxes and click Save.

After you have secured this resource, you may need to repeat the process for a lower-level resource. The YAB Resource BOM tool returns two levels of dependencies. If a dependency of the dependent resource you just secured is also not secure, you must start again and insert the menu label you just secured to find its associated dependencies.

Examples

Browses

This example demonstrates how to determine which lookups are necessary for the Sales Quotes browse. You start by using **sec-resource-uri** to determine the browse URI:

```
> yab sec-resource-uri -type:browse -label:"Sales Quotes"
Label        URI
------------ ----------------------
Sales Quotes urn:browse:mfg:qo801
Sales Quotes urn:browse:mfg:sq001
```

Next, look at the **sec-resource-bom** output for these resources:

```
$ yab sec-resource-bom urn:browse:mfg:qo801 superuser
L Label              URI                  Perm
- ------------------ -------------------- ----
0 Sales Quotes       urn:browse:mfg:qo801 R
1 Lookups
2 Sites              urn:browse:mfg:gp348 R
2 Salesperson Master urn:browse:mfg:gp245 R
2 MISSING RESOURCE   urn:browse:mfg:gp206
2 MISSING RESOURCE   urn:browse:mfg:gp012
2 Customer Master    urn:browse:mfg:gp317 R
2 Language Master    urn:browse:mfg:gp158 R
2 Channel Codes      urn:browse:mfg:gp392 R
2 Currency Master    urn:browse:mfg:mc005 R
1 Drilldowns
2 Language Master    urn:browse:mfg:gp158 R
2 Customer Master    urn:browse:mfg:gp317 R
2 MISSING RESOURCE   urn:browse:mfg:gp012
2 MISSING RESOURCE   urn:browse:mfg:gp206
2 Currency Master    urn:browse:mfg:mc005 R
2 Salesperson Master urn:browse:mfg:gp245 R
2 Sites              urn:browse:mfg:gp348 R


$ yab sec-resource-bom urn:browse:mfg:sq001 superuser
L Label            URI                  Perm
- ---------------- -------------------- ----
0 Sales Quotes     urn:browse:mfg:sq001 R
1 Lookups
2 Item Master      urn:browse:mfg:gp340 R
2 Customer Master  urn:browse:mfg:gp317 R
2 MISSING RESOURCE urn:browse:mfg:gp206
1 Drilldowns
2 MISSING RESOURCE urn:browse:mfg:gp206
2 Customer Master  urn:browse:mfg:gp317 R
2 Item Master      urn:browse:mfg:gp340 R
2 MISSING RESOURCE urn:browse:mfg:ad001
```

This output provides all of the lookups and drilldowns that are associated with the browses. If the user needs full access to the browse, then permissions for all of these associated browses must be granted. Some of the returned lookups have a “MISSING RESOURCE” label. This indicates that

the lookup or drilldown is referenced in the browse, but is not defined in the system. These missing resources must be defined in the `browseresources.xml` file for the appropriate module before permissions can be set.

Commodity Code

This example demonstrates how to determine the resources necessary to use the Commodity Code menu item. You start by using the **sec-resource-uri** process to determine the URI for the menu item. You can filter by label and resource type because you only need view resources. The output shows you the URI for the Commodity Code and the Commodity Code Detail views. The highlighted URI in Figure 4.10 is for the main Commodity Code menu item, which is the information you require.

Fig. 4.10
Commodity Code URI

```
$ yab sec-resource-uri -label:"Commodity Code" -type:view
Label                URI
-------------------- ------------------------------------------------------------
COMMODITY_CODE       urn:view:hybridbrowse:com.qad.erp.base.commodityCodeMasters
COMMODITY_CODE_DETAIL urn:view:hybridbrowse:com.qad.erp.base.commodityCodeDetails
```

Next, use **sec-resource-bom** to determine dependencies and missing permissions for the CodeMaster role. The BOM, shown in Figure 4.11, shows you that the Commodity Code view has dependencies on the ICommodityCodeDetail business entity, the Commodity Codes browse, and the Units of Measure lookup, all highlighted in red. It also shows you that the CodeMaster role does not have permissions to access the business entity and lookup resources because the Perm column is blank for those two resources.

Fig. 4.11
CodeMaster Role BOM

```
$ yab sec-resource-bom urn:view:hybridbrowse:com.qad.erp.base.commodityCodeMasters CodeMaster

L Label                                URI                                                          Perm
- ------------------------------------- ------------------------------------------------------------ ----
0 COMMODITY_CODE                       urn:view:hybridbrowse:com.qad.erp.base.commodityCodeMasters R
1 Dependent Entities
2 ICommodityCodeDetail [no label defined] urn:be:com.qad.base.item.ICommodityCodeDetail             BE with no permission
2 Commodity Code                       urn:be:com.qad.base.item.ICommodityCodeMaster                CDRW
1 Browses
2 Commodity Codes                      urn:browse:mfg:tx003                                         Browse with Read permission  R
1 Maintenance Lookups
2 Units of Measure                     urn:browse:mfg:gp276                                         Maintenance Lookup with no permission
```

You need to assign the CodeMaster role the appropriate permissions for both the business entity and the lookup in Role Permissions Maintain. See “Assigning Permissions to Roles” on page 25 for information on assigning permissions. If you only grant Read access to ICommodityCodeDetail, the CodeMaster role can see data but not change it. Granting full permissions of Create, Delete, Read, and Write enables the users assigned that role to both see and make changes to the Commodity Code details. You also need to give Read permissions for the Units of Measure lookup. After you set these permissions, the user can access the Commodity Code maintenance screen and make changes.

After making these updates, the user may see “Error 403: Access Denied” when trying to add a new Commodity Code Detail Line in the Item Number lookup because that lookup is linked with Commodity Code Detail and not Commodity Code Master. You must run the tool again for Commodity Code Detail to determine the lower-level dependencies and then grant proper permissions.

Fig. 4.12
Commodity Code Detail

```
$ yab sec-resource-bom urn:view:hybridbrowse:com.qad.erp.base.commodityCodeDetails CodeMaster
L Label                                 URI                                                          Perm
- ------------------------------------ ------------------------------------------------------------ ----
0 COMMODITY_CODE_DETAIL                 urn:view:hybridbrowse:com.qad.erp.base.commodityCodeDetails R
1 Dependent Entities
2 ICommodityCodeDetail [no label defined] urn:be:com.qad.base.item.ICommodityCodeDetail              CDRW
1 Browses
2 Commodity Code Detail                 urn:browse:mfg:tx014                                         R
1 Browse Lookups
2 Item Descriptions                     urn:browse:mfg:gp197
1 Drilldowns
2 Item Descriptions                     urn:browse:mfg:gp197
1 Maintenance Lookups
2 Item Master                           urn:browse:mfg:gp340
```

Commodity Code Detail has a dependent entity and a browse with full permissions already assigned, but you still must grant permission to a browse lookup, maintenance lookup, and drilldown.

Note After granting permissions, you may still see an Access Denied pop-up window when you select an item from a lookup. This is due to a Javascript event handler that makes calls to the Items REST API. You can still do CDRW operations on the Commodity Code master and detail records. See “Limitations” on page 42 for more information.

Sales Order

This example demonstrates how to determine the resources necessary to use the Sales Order menu item. As with Commodity Code, you start with the **sec-resource-uri** process to determine the URI for the menu item. Sales Order returns more possibilities than Commodity Code. You need to look at the label items to find the one that matches your menu item, highlighted in red.

Fig. 4.13
Sales Order

```
$ yab sec-resource-uri -label:"Sales Order" -type:view
Label                                URI
------------------------------------ ------------------------------------------------------------------
Quotes not converted to Sales Orders urn:view:browse:com.qad.erp.sales.quotesNotConvertedToOrdersBrowse
SALES_ORDERS                         urn:view:hybridbrowse:com.qad.erp.sales.salesOrders
SALES_ORDER_LINES                    urn:view:hybridbrowse:com.qad.erp.sales.salesOrderLines
Sales Order Credit                   urn:view:hybridbrowse:com.qad.erp.sales.salesOrderCredits
Sales Order Detail                   urn:view:browse:com.qad.erp.sales.salesOrderDetailBrowse
Sales Order Gross Margin             urn:view:browse:com.qad.erp.sales.salesOrderGrossMarginBrowse
Sales Order Price                    urn:view:browse:com.qad.erp.sales.salesOrderPriceBrowse
Sales Orders to Ship                 urn:view:browse:com.qad.erp.sales.salesOrdersToShipBrowse
```

Sales Order URI for resource BOM tool

After you find the matching label, copy its URI to use in the resource BOM. Next, use **sec-resource-bom** to determine dependencies and missing permissions for the SalesRep role.

Fig. 4.14
Sales Rep Role Output

```
 $ yab sec-resource-bom urn:view:hybridbrowse:com.qad.erp.sales.salesOrders SalesRep
L Label                                  URI                                                  Perm
- ---------------------------------------- ---------------------------------------------------- ----
0 SALES_ORDERS                           urn:view:hybridbrowse:com.qad.erp.sales.salesOrders R
1 Dependent Entities
2 Sales Order                            urn:be:com.qad.sales.salesorder.ISalesOrderHeader   CDRW
2 ISalesOrderComment [no label defined]  urn:be:com.qad.sales.salesorder.ISalesOrderComment  CDRW
2 Sales Order Line                       urn:be:com.qad.sales.salesorder.ISalesOrderLine     CDRW
2 com.qad.sales.salesorder.ISalesOrderTaxes urn:be:com.qad.sales.salesorder.ISalesOrderTaxes
1 Browses
2 Customer Sales Orders                  urn:browse:mfg:so803                                R
1 Browse Lookups
2 Sales Order Master                     urn:browse:mfg:gp239
2 Sales Order Master by PO               urn:browse:mfg:gp242
2 Daybook Set Codes                      urn:browse:mfg:so036
2 Language Master                        urn:browse:mfg:gp158
2 Currency Master                        urn:browse:mfg:mc005
2 Customer Master                        urn:browse:mfg:gp317
2 Customer Master                        urn:browse:mfg:gp317
2 Channel Codes                          urn:browse:mfg:gp392
2 MISSING RESOURCE                       urn:browse:mfg:gp255
2 MISSING RESOURCE                       urn:browse:mfg:gp306
2 Salesperson Master                     urn:browse:mfg:gp245
2 Action Statuses                        urn:browse:mfg:gp005
2 MISSING RESOURCE                       urn:browse:mfg:gp012
2 Sites                                  urn:browse:mfg:gp348
2 MISSING RESOURCE                       urn:browse:mfg:gp243
1 Drilldowns
2 Code Master                            urn:browse:mfg:gp072
2 Customer Master                        urn:browse:mfg:gp317
2 Customer Master                        urn:browse:mfg:gp317
2 Currency Master                        urn:browse:mfg:mc005
2 MISSING RESOURCE                       urn:browse:mfg:gp255
2 MISSING RESOURCE                       urn:browse:mfg:gp012
2 Salesperson Master                     urn:browse:mfg:gp245
2 Sales Order Master                     urn:browse:mfg:gp239
2 Sites                                  urn:browse:mfg:gp348
2 Language Master                        urn:browse:mfg:gp158
2 MISSING RESOURCE                       urn:browse:mfg:gp243
2 Sales Order Master by PO               urn:browse:mfg:gp242
1 Maintenance Lookups
2 Delivery Terms                         urn:browse:mfg:gp320
2 Action Statuses                        urn:browse:mfg:gp005
```

This shows that the Sales Order screen depends on the SalesOrderComment, SalesOrderLine, and SalesOrderTaxes business entities, as well as multiple lookups and drilldowns. The business entities are highlighted in Figure 4.15.

Fig. 4.15
Sales Rep Role Dependencies

```
 $ yab sec-resource-bom urn:view:hybridbrowse:com.qad.erp.sales.salesOrders SalesRep
L Label                                  URI                                                  Perm
- ---------------------------------------- ---------------------------------------------------- ----
0 SALES_ORDERS                           urn:view:hybridbrowse:com.qad.erp.sales.salesOrders R
1 Dependent Entities
2 Sales Order                            urn:be:com.qad.sales.salesorder.ISalesOrderHeader   CDRW
2 ISalesOrderComment [no label defined]  urn:be:com.qad.sales.salesorder.ISalesOrderComment  CDRW
2 Sales Order Line                       urn:be:com.qad.sales.salesorder.ISalesOrderLine     CDRW
2 com.qad.sales.salesorder.ISalesOrderTaxes urn:be:com.qad.sales.salesorder.ISalesOrderTaxes
1 Browses
2 Customer Sales Orders                  urn:browse:mfg:so803                                R
```

Full permissions have been granted for the business entities, but you must grant the role permissions for the lookups, browses, and drilldowns.

Next, you need to determine the dependencies for Sales Order Line.

Fig. 4.16
Sales Order Line Dependencies

```
 yab sec-resource-bom urn:view:hybridbrowse:com.qad.erp.sales.salesOrderLines SalesRep
L Label                                 URI                                                          Perm
- ------------------------------------- ------------------------------------------------------------ ----
0 SALES_ORDER_LINES                     urn:view:hybridbrowse:com.qad.erp.sales.salesOrderLines R
1 Dependent Entities
2 ISalesOrderLineComment [no label defined] urn:be:com.qad.sales.salesorder.ISalesOrderLineComment  CDRW
2 Sales Order Line                      urn:be:com.qad.sales.salesorder.ISalesOrderLine         CDRW
1 Browses
2 Sales Order Detail                    urn:browse:mfg:so804                                         R
1 Browse Lookups
2 Code Master                           urn:browse:mfg:gp072
2 Units of Measure                      urn:browse:mfg:gp358
2 Item Master                           urn:browse:mfg:gp340
2 Sites                                 urn:browse:mfg:gp348
2 Sales Order Master                    urn:browse:mfg:gp239
2 MISSING RESOURCE                      urn:browse:mfg:gp396
2 Item Descriptions                     urn:browse:mfg:gp197
2 Daybook Set Codes                     urn:browse:mfg:so036
2 MISSING RESOURCE                      urn:browse:mfg:gp255
2 MISSING RESOURCE                      urn:browse:mfg:gp306
1 Drilldowns
2 Sales Order Master                    urn:browse:mfg:gp239
2 Item Descriptions                     urn:browse:mfg:gp197
2 Sites                                 urn:browse:mfg:gp348
2 Item Master                           urn:browse:mfg:gp340
2 MISSING RESOURCE                      urn:browse:mfg:gp255
2 Units of Measure                      urn:browse:mfg:gp358
2 Code Master                           urn:browse:mfg:gp072
1 Maintenance Lookups
2 Service Warranty Types                urn:browse:mfg:gp260
2 Customer/Ship-Tos                     urn:browse:mfg:gp087
2 Delivery Terms                        urn:browse:mfg:gp320
2 Code Master                           urn:browse:mfg:gp072
2 Sales Order Detail                    urn:browse:mfg:gp241
2 Freight Classes                       urn:browse:mfg:so001
2 Valid Cost Centers                    urn:browse:mfg:gp600
2 Trailer Codes                         urn:browse:mfg:gp265
2 EMT Types                             urn:browse:mfg:gp321
2 Features                              urn:browse:mfg:so078
2 Item Lot/Serials                      urn:browse:mfg:so077
```

Cross-reference this list of lookups with the Sales Order screen to determine which lookups are required for access.

Limitations

- Granularity of results – While a view may require access to a business entity (BE), it does not necessarily need access to every field in that BE. Thus, it may be possible to assign permissions at a more granular level than the BE, and still get the required results.
- Excess lookups – The tool currently returns all lookups associated with a BE that is associated with a view. A view, however, may use only a subset of the lookups, so you may get back more than you need.
- API calls in Javascript Event Handlers – The tool only works with dependencies that are defined in View Metadata, View Resource Metadata, and Entity Metadata. However, some modules have additional rest API calls made from Javascript. The current tool does not have the capability to find all missing resource permissions.

Using Tomcat Logs

All authorization failures are logged on the Channel Islands UI Tomcat server under `servers/tomcat-webui/logs/catalina.out` as follows:

```
[15/12/17@13:51:07.162-0800] WARN
com.qad.qracore.security.authorization.QraPermissionEvaluator: Requested
permission READ is denied for User qad for Resource
urn:service:com.qad.distribution.logisticsaccounting.ILogisticsAcctControl
```

To use the Tomcat log to identify missing permissions:

1. Log into the Channel Islands UI as the user that has insufficient permissions.
2. Use the functionality on the screen to which you want to grant access.
3. Examine the log file using the following:

   ```
   $ grep "QraPermissionEvaluator.*denied.*USER" servers/tomcat-
   webui/logs/catalina.out
   ```

   where USER is the user name that has insufficient permissions; for example, for the qad user:

   ```
   $ grep "QraPermissionEvaluator.*denied.*qad" servers/tomcat-
   webui/logs/catalina.out
   ```

4. This returns all resources for which the user was denied access. Use the timestamps on the log entries to determine which resources were denied when trying to access the desired menu.

Identifying Resources by URI

Table 4.2 provides an overview of the types of resources and how they are uniquely identified in the system. These unique identifiers are used for permissions to reference the associated resources. They are also required for authorization implementation

Table 4.2 Identification of Resources by URI

Description	Resource Classification	Resource Type	Resource Identifier	Resource URI
System Modules Container		container	module	urn:container:module
QRA Sales Module	module	module	com.qad.sales	urn:module:com.qad.sales
QRA Sales Module Views Container		container:view	module:com.qad.sales	urn:container:view:module:com.qad.sales
QRA Sales Module Business Entities Container		container:be	module:com.qad.sales	urn:container:be:module:com.qad.sales
QRA Sales Order Business Entity	businessentity	be	com.qad.sales.salesorder.ISalesOrder	urn:be:com.qad.sales.salesorder.ISalesOrder
QRA Sales Order Business Entity Field Groups Container		container:fg	be:com.qad.sales.salesorder.ISalesOrder	urn:container:fg:be:com.qad.sales.salesorder.ISalesOrder
QRA Sales Order Business Entity Views Container		container:view	be:com.qad.sales.salesorder.ISalesOrder	urn:container:view:be:com.qad.sales.salesorder.ISalesOrder
QRA Sales Order Field Group “main”	fieldgroup	fg	com.qad.sales.salesorder.ISalesOrder:main	urn:fg:com.qad.sales.salesorder.ISalesOrder:main
QRA Sales Order Field “Note”	field	field	com.qad.sales.salesorder.ISalesOrder:SalesOrderLine.Note	urn:field:com.qad.sales.salesorder.ISalesOrder:SalesOrderLine.Note
QRA Sales Order browse (datasource, not view)	browse	browse:qra	com.qad.sales.salesorder.ISalesOrder	urn:browse:qra:com.qad.sales.salesorder.ISalesOrder
Sales Order Hybrid Browse View	view	view		urn:view:hybridbrowse:com.qad.sales.salesorder
Sales Order Standalone Browse View	view	view		urn:view:browse:com.qad.sales.salesorder
MFG Browse datasource	browse	browse:mfg	so003	urn:browse:mfg:so003
FIN Browse datasource	browse	browse:fin	bgl.selectgl	urn:browse:fin:bgl.selectgl
MFG/PRO Sales Order	program	program:sosomt		urn:program:sosomt
Fin Supplier Invoice	businesscomponent	businesscomponent:bcinvoice		urn:businesscomponent:bcinvoice