WebSphere Data Interchange Version 3 Release 2

Security in the WDI V3.2 Client Implementation Guide

May 24, 2005
Revised July 6, 2006

Version 2.1

Revision History

Date	Version	Description	Author
18 Feb 2005	1.0	Initial draft	Richard Bennett
25 Feb 2005	1.0.1	Updated draft	David Hixon
18 Mar 2005	1.0.2	Edited wording	Robin Pope
12 May 2005	1.0.3	Added Tables - pages 6-8	Bill Whitehead
24 May 2005	1.0.4	Added table EDIMAPCSTDET	Bill Whitehead
6 Jan 2006	2.0	Added instructions on restricting which rows a user can see	David Hixon
6 July 2006	2.1	Updated TS views and PSBI	Robin Pope

Table of Contents

Security in the WebSphere Data Interchange V3.2.1 Client

The ability to control the access of users according to role is a repeatedly requested customer requirement for the WDI V3.2 Client. Customers need the ability to restrict the components of the WDI Client to a set of users. This is also called Role Based Access Control, or RBAC. Typical roles are Mapping, Trading Partner Maintenance, Configuration or Setup data Administration, and Operations (e.g. users of Transaction Store and Event Log).

Role	Function
Mapper	a person who uses Data Formats, Standards, XML Schemas to build a Map;
EDIAdmin	a person who configures WDI for a new Trading Partner, installs new Standard Transactions,
SystemsIntegrator	a person who implements a procedural changes, i.e. changes to data flows and MQSeries.
SysAdmin	a person who administers permissions to the database users.
Operator	a person who reviews translated data for errors and uses the client for problem determination of translations.

In a typical EDI Installation, the roles that people play should determine the database access they are granted. To give access outside of the role domain is to unnecessarily expose the system to inadvertent or malicious changes. The current WDI Client only allows a user to access tables in update mode. This occurs because the WDI Client attempts to “lock” any object it is opening. The locking mechanism requires update access to the object. If a person is not granted update access to an object, then DB2 security prohibits access to the object.

With this new WDI V3.2 CSD, a “View” functionality is implemented in conjunction with a set of DB2 GRANTS to implement a role based security capability. “View” works exactly like “Open” except that the editors and related dialogs are opened in “read-only” mode. The “read-only” mode is not new to the Client. The Client has supported read-only mode for all editors and related dialogs in the past. This applied when an object that was “locked” was opened. The client displayed the editor in read-only mode. The “View” function simply provides a mechanism for the user to open the editors and related dialogs in read-only mode without the object being locked. The “View” function is available on the “File” menu beneath the “Open” function.

The preferences dialog has been updated to allow the user to indicate whether “Open” or “View” will be the default when the current “Open” button is pressed on the toolbar. The setting of this preference will also affect the action that occurs when the user double clicks on an object that normally would produce an “Open” action. This preference setting allows users that only have read access to the database to work efficiently without the added steps of going to the File menu. Access to tables can then be controlled using database authorizations (GRANT) and views.

The tooltip that displays above the toolbar will reflect the user’s preference.

The following additional related changes will be available:

- Any editor that contains an "Edit" button used to open a referenced object (these are located in the data format and standards editors) will change the label on the button to "View" when the editor is opened as "View".
- Any editor that contains a tree or list that provides an "Open" function in a popup menu will now also provide a "View" function.
- Ctrl+V is implemented as a hotkey to invoke the "View" function.
- "View" and "Open" perform the same function for the Message Log, Transaction Store, and Event Log. These objects have always been “read-only”.

A number of sets of DB2 Data Definition Language (DDL) are provided to show how different role based access scenarios can be implemented. Review the Conceptual Model section of this document to understand the relationship between “permissions”, WDI DB2 Tables, and roles.

Role Based Access Scenarios

Five possible scenarios that would use this role based access control implementation are described in this section. The first four scenarios describe situations where it is required that access to configuration artifacts be controlled based upon the **user**, and the **type** of artifact (trading partners, maps, etc.) that the user wants to access. In the fifth scenario, access is controlled based upon the user, the type of artifact and the **members** of the artifact. This last scenario demonstrates how to configure WDI and DB2 such that a user is restricted to a subset of an artifact, for example just those trading partners in North America.

1) A customer wants to restrict the Mapping staff from updating Trading Partner Information. Mappers should be able to change Data Formats, Standards, XML definitions, code lists, and Maps. They could view, but should not be able to change, Setup data (profiles). The should not be allowed to view Transaction Store.

   Solution

   A. Create a group called "MAPPERS" using the access control tool on the platform where the WDI database resides. For example, on z/OS you could use RACF to create the group.
   B. Add the user ID's of the Mapping Staff to the group "MAPPERS".
   C. Copy the file *mapper_perms.ddl* to a name of your choosing.
   D. Review the Mapper Class section and determine to which configuration artifacts "mappers" should be given access in your installation.
   E. Change the "SELECT, INSERT, UPDATE, DELETE" clauses as required on the objects to which you need to restrict access. To prevent any access to a configuration artifact, comment out the appropriate GRANT statements. In this case, the Transaction Store objects should be commented out.
   F. Do a global change from "User01" to "GROUP MAPPERS"
   G. Submit the DDL file to DB2 and verify that it ran successfully.

2) The Implementation Group of a customer can migrate and maintain Usages, maintain Trading Partners, and view Transaction Store and the Event Log. They can view but not change Maps and mapping objects.

   Solution

   A. Create a group called "IMPLMNTS" using the access control tool on the platform where the WDI database resides. For example, on z/OS you could use RACF to create the group.
   B. Add the user ID's of the Implementation Group to the group "IMPLMNTS".

C. Copy the file *EDIAdmin_perms.ddl* to a name of your choosing.

D. Review the EDIAdminr Class section and determine to which configuration artifacts "administrators" should be given access in your installation.

E. Change the "SELECT, INSERT, UPDATE, DELETE" clauses as required on the objects to which you need to restrict access. To prevent any access to a configuration artifact, comment out the appropriate GRANT statements.

F. Do a global change from "User01" to "GROUP IMPLMNTS"

G. Submit the DDL file to DB2 and verify that it ran successfully.

3) A special group has been asked to maintain Code Lists. They can view Standards and nothing else. They need the ability to add, update, delete entries in a Code List – including the creation of new Code Lists.

Solution

A. Create a group called "CODELIST" using the access control tool on the platform where the WDI database resides. For example, on z/OS you could use RACF to create the group.

B. Add the user ID's of the special group to the group "CODELIST".

C. Copy the file *EDIAdmin_perms.ddl* to a name of your choosing.

D. Review the EDIAdmin Class section and determine to which configuration artifacts "code list" should be given access in your installation.

E. Change the "SELECT, INSERT, UPDATE, DELETE" clauses as required on the objects to which you need to restrict access. To prevent any access to a configuration artifact, comment out the appropriate GRANT statements. In this case, comment out all the GRANT statements except those under the heading "Standards Tables".

F. Do a global change from "User01" to "GROUP CODELIST"

G. Submit the DDL file to DB2 and verify that it ran successfully.

H. Repeat the last 2 steps for each userid in the "cdoe list" group

4) The Operations group has view only access to all components in the WDI Client.

Solution

A. Create a group called "OPERATOR" using the access control tool on the platform where the WDI database resides. For example, on z/OS you could use RACF to create the group.

B. Add the user ID's of the Operations Group to the group "OPERATOR".

C. Copy the file *operator_perms.ddl* to a name of your choosing.

D. Review the Operator Class section and determine to which configuration artifacts "operators" should be given access in your installation.

E. Change the "SELECT, INSERT, UPDATE, DELETE" clauses as required on the objects to which you need to restrict access. To prevent any access to a configuration artifact, comment out the appropriate GRANT statements.

F. Do a global change from "User01" to "GROUP OPERATOR"

G. Submit the DDL file to DB2 and verify that it ran successfully.

5) A global company wants to consolidate all their EDI processing to a single data center and a single production database. They do not want to consolidate the EDI staffs into one staff because time zones, local customs and language barriers make it difficult to deal with trading partners all over the world from a single location. They are concerned that the EDI staff in one region might unintentionally damage the configuration for another region, so they would like to restrict the access of each of the regional offices to just those configuration artifacts that apply to them. For example, the Asia-Pacific region users should not be able to see or modify the trading partners for the North American and European regions, and vice versa.

A. Copy the file *views.ddl* three times to names of your choosing. For this scenario we assume the names are *asia_config.ddl*, *namerica_config.ddl*, and *europe_config.ddl*.

B. Edit the file *asia_config.ddl* and make the following global changes:

1. From "ROLE." to "ASIA.".
2. from "no_access_list" to "'NA, EU'"

C. Submit the DDL file to DB2 and verify that it ran successfully.

D. Edit the file *na_config.ddl* and make the following global changes:

1. From "ROLE." to "NAMERICA.".
2. from "no_access_list" to "'AS, EU'"

E. Submit the DDL file to DB2 and verify that it ran successfully.

F. Edit the file *na_config.ddl* and make the following global changes:

1. From "ROLE." to "EUROPE.".
2. from "no_access_list" to "'AS, NA'"

G. Submit the DDL file to DB2 and verify that it ran successfully.

H. Create a group called "ASIA", a group called "NAMERICA" and a group called "EUROPE" using the access control tool on the platform where the

WDI database resides. For example, on z/OS you could use RACF to create the group.

I. Add the user ID's of the EDI staff in the Asia-Pacific region to the group "ASIA", the user ID's of the EDI staff in the North American region to the group "NAMERICA" and the user ID's of the EDI staff in the North American region to the group "EUROPE".

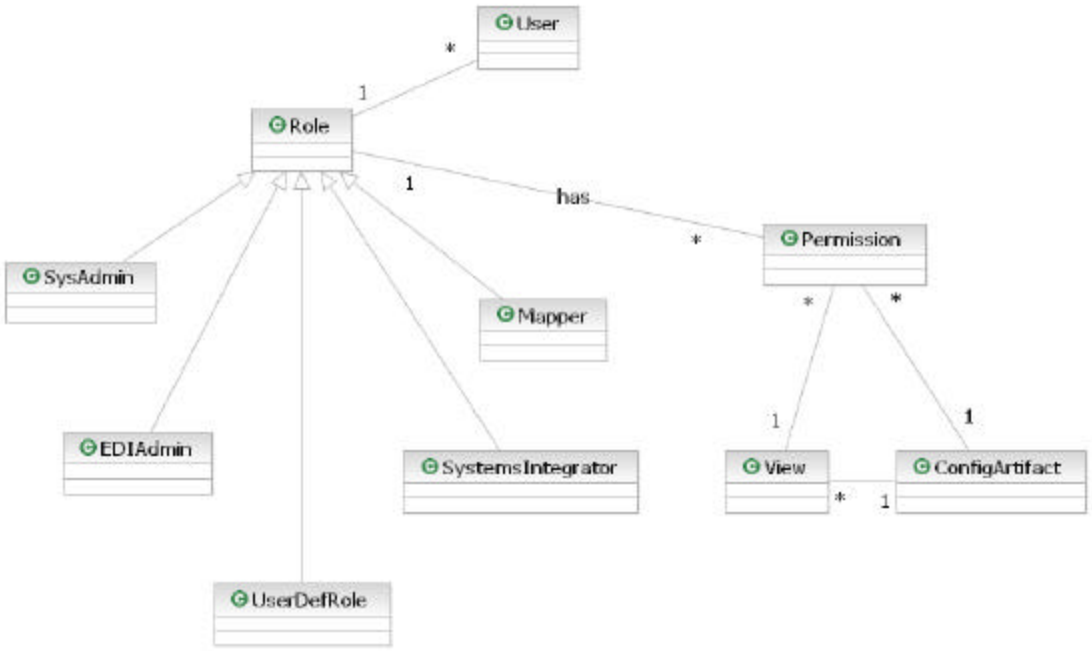
J. For each region (Europe, North America and Asia):

   1. Determine the set of roles the region requires, such as mapper, operator, system adminstrator, etc.
   2. Copy the appropriate permission files (such as *sysadmin_perms.ddl* and *mapper_perms.ddl)* to names of your choosing, for example *europe_mapper.ddl*.
   3. Change the "SELECT, INSERT, UPDATE, DELETE" clauses as required on the objects to which you need to restrict access. To prevent any access to a configuration artifact, comment out the appropriate GRANT statements.
   4. Do a global change from "User01" to "GROUP *region_name*", where *region_name* is either ASIA, NAMERICA, or EUROPE.
   5. Submit the DDL file to DB2 and verify that it ran successfully.

K. For each client system in each region

   1. Create a new system and name it the region name. Specify the same database parameters for the new system as the base EDIENU32 system, except for the "Database Qualifier" field. Enter the region name in that field.
   2. Tell the tell the client system user to use the system corresponding to their region name and role when connecting to the consolidated database.

Conceptual Model



User

A “User” is a user of the WDI client tooling. The implementation of a user in 3.2 is a user in whatever security facility the customer uses on the system that contains the WDI database that will be accessed. On the z/OS platform this would typically be RACF.

Role

A “Role” is basically a collection of “Permissions”. A role has one permission for every type of configuration artifact in WDI. Users can fulfill one or more roles for the product. There are five pre-defined roles: “SysAdmin” who manages users and permissions, “EDIAdmin” who manages the trading partner community, “SystemsIntegrator” who does things like set up queues and service profiles to configure the flow of messages through the product, “Mapper” who manages the maps that transform messages, and “Operator” who handles customer data exceptions and problems. In addition, users can create their own user-defined roles as well, with custom sets of permissions and views.

The implementation of a role in WDI 3.2 is as a group within whatever security facility the customer uses on the system that contains the WDI database that will be accessed. On the z/OS platform this would typically be RACF.

Permission

There are permissions for each type of configuration artifact in WDI. There are four types of access that a permission can grant:

- *Read* – the user is allowed to bring up an instance of the artifact in a read-only editor.
- *Insert* – The user is allowed to create new instances of the artifact.
- *Update* – The user is allowed to update instances of the artifact.
- *Delete* – The user is allowed to delete instances of the artifact.

In addition to the type of access allowed, there can be a "View" associated with the permission. The view defines a subset of instances of the configuration artifact that can be seen by the users that fulfill the role associated with the permission.

The implementation of a permission in WDI 3.2 is a SQL GRANT statement. The "SELECT, INSERT, UPDATE, DELETE" clause on the GRANT statement corresponds to the Read, Insert, Update and Delete access permissions allowed in WDI. There is a file called *role_perms.ddl* provided with the product that the user can copy, rename and edit as required to create permission sets with restricted access to some or all of the configuration artifacts. Access to a particular configuration artifact is accomplished by changing the "SELECT, INSERT, UPDATE, DELETE" clause on the corresponding tables. To prevent any access to a configuration artifact, the user should comment out or delete the GRANT statement for the artifact.

Configuration Artifact

A configuration artifact is an object within the WDI client that can be edited or accessed via an editor. The implementation of the ConfigArtifact class in WDI 3.2 is the set of DB2 tables used to represent the object in the WDI database.

The configuration artifacts are:

Description	WDI DB2 Table name
Activity Log	EDIENU32.EDIPSAC
Application Defaults Profile	EDIENU32.EDIPSAP
CICS Performance Profile	EDIENU32.EDIPSSY
Continuous Receive Profile	EDIENU32.EDIPSCR
E Envelope Profile	EDIENU32.EDIPSEE
I Envelope Profile	EDIENU32.EDIPSIE
Language Profile	EDIENU32.EDIPSLP
Mailbox Profile	EDIENU32.EDIPSRQ

Description	WDI DB2 Table name
MCD Profile	EDIENU32.EDIPSMCD
MQ Series Queue Profile	EDIENU32.EDIPSMQ
Network Command Profile	EDIENU32.EDIPSNO
Network Profile	EDIENU32.EDIPSNP
Network Security Profile	EDIENU32.EDIPSSP
Service Profile	EDIENU32.EDIPSSL
T Envelope Profile	EDIENU32.EDIPSTE
U Envelope Profile	EDIENU32.EDIPSUE
User Exits Profile	EDIENU32.EDIPSAD
X Envelope Profile	EDIENU32.EDIPSXE
Trading Partner	EDIENU32.EDIPROF EDIENU32.EDIPSTP EDIENU32.EDIPSBI
Contact	EDIENU32.EDITPCM EDIENU32.EDITPCN EDIENU32.EDITPCT
Data Transformation Map Rule	EDIENU32.EDIRULE
Send Map Usage	EDIENU32.EDITPST
Receive Map Usage	EDIENU32.EDITPRT
Map Application Control Fields	EDIENU32.EDIMAPAPPLCNTL
Mapping Commands	EDIENU32.EDIMAPCMDS
Mapping Elements	EDIENU32.EDIMAPELE
Map Global Variables	EDIENU32.EDIMAPGBLVAR
Map Header	EDIENU32.EDIMAPHEAD
Map Local Variables	EDIENU32.EDIMAPLCLVAR
Mapping Command Nodes	EDIENU32.EDIMAPNODES
Mapping Cross References	EDIENU32.EDIMAPREF
Mapping Segments	EDIENU32.EDIMAPSEG
Mapping Syntax	EDIENU32.EDIMAPSYNTAX
Map Control String	EDIENU32.EDICSTX EDIENU32.EDIMAPCSTHDR EDIENU32.EDIMAPCSTDET
Map Report	EDIENU32.EDIMAPRPT
Data Format Dictionary	EDIENU32.EDIADFDICT
Data Format Record ID information	EDIENU32.EDIADFRECIDINFO
Data Format	EDIENU32.EDIADFHEADER EDIENU32.EDIADFHDRMEM

Description	WDI DB2 Table name
Data Format Loop	EDIENU32.EDIADFLOOP EDIENU32.EDIADFLOOPMEM
Data Format Record	EDIENU32.EDIADFRECORD EDIENU32.EDIADFRECMEM
Data Format Structure	EDIENU32.EDIADFSTRUCT EDIENU32.EDIADFSTRUCTMEM
Data Format Field	EDIENU32.EDIADFFIELD
EDI Standard Dictionary	EDIENU32.EDISTDSTH
EDI Standard Transaction	EDIENU32.EDISTDTXD EDIENU32.EDISTDTXH EDIENU32.EDISTDTXN
EDI Standard Segment	EDIENU32.EDISTDSGD EDIENU32.EDISTDSGH EDIENU32.EDISTDSGN
EDI Standard Data Element	EDIENU32.EDISTDDED EDIENU32.EDISTDDEH
EDI Standard Composite Data Element Notes	EDIENU32.EDISTDCDN
EDI Standard Code List	EDIENU32.EDIPSTV EDIENU32.EDIPSTT EDIENU32.EDIPSTD
EDI Envelope Standard	EDIENU32.EDISTDENV
EDI Envelope Control String	EDIENU32.EDICSTX EDIENU32.EDIMAPCSTHDR EDIENU32.EDIMAPCSTDET
XML Dictionary	EDIENU32.EDIXMLDICT
XML DTD	EDIENU32.EDIDTDHDR EDIENU32.EDIDTD
XML Schema	EDIENU32.EDIDTDHDR EDIENU32.EDIDTD
XML Namespace	EDIENU32.EDIXMLNS
Transaction Store – All objects	EDIENU32.EDITSTH EDIENU32.EDITSEV EDIENU32.EDITSGP EDIENU32.EDITSTI EDIENU32.EDITSTO EDIENU32.EDITSAU EDIENU32.EDITSLT EDIENU32.EDIVTSTH

Description	WDI DB2 Table name
	EDIENU32.EDIVTSIS
Event Log	EDIENU32.EDIELOG
Management Reporting Tables	EDIENU32.EDIMRCM EDIENU32.EDIMRPC EDIENU32.EDIMRPR EDIENU32.EDIMRPS EDIENU32.EDIMRRT EDIENU32.EDIMRST

Views

A “View” defines a subset of instances of the configuration artifact that can be seen by the users. It is used to limit which instances of a particular configuration artifact that the user can affect. For example, if a user should be allowed to access the X12 standards, but not any other kind of standard, then a view could be created that shows only standards beginning with the three characters “X12” - assuming that all X12 standards begin with X12..... A view can be created on any configuration artifact.

The implementation of the View class in WDI 3.2 is a DB2 view. There is a file called *views.ddl* provided with the product that the user can copy, rename and edit as required to create views. In the file there is a “CREATE VIEW role_name.table_name AS SELECT * FROM EDIENU32.table_name” statement for each table in the WDI database.

Example

Restrict a mapper role to only be able to see rows defining X12 standards, the following CREATE VIEW statement for the EDISTDSTH table could be used:

```
--Create a role specific view of the Standard Dictionary table;
  CREATE VIEW MAPPER.EDISTDSTH AS SELECT * FROM
      EDIENU32.EDISTDSTH
      WHERE STDID = 'X12%'
--Terminate the CREATE VIEW statement:
 ;
--Create a VIEW  for every table. For those that do not need to be restricted, the WHERE clause is not needed.
```

The system administrator is able to restrict the instances of each configuration artifact that a particular role sees by adding an appropriate WHERE clause to the views that are used to represent the configuration artifact.

Groups

A Group is a collection of users who have the same role. An alternative to defining each user with a unique set of authorizations is to assign the authorizations to a GROUP and then assign individual users to that GROUP. This technique is not available on all platforms.

With DB2, the technique is shown below:

.

```
--Grant authority for the group "role_name" to access this
--table by changing the following grant statment as
--required:
  GRANT SELECT, INSERT, DELETE, UPDATE ON role_name.EDISTDSTH
    TO GROUP role_name;
```

On z/OS, then add users to the RACF Group name “role_name”.
On Windows Server, use Windows Users and Passwords Administration to associate users with the Group.
On AIX, use SMIT or SMITTY to associate users with the Group.

On Windows 2000 Professional and Windows XP, use DB2 Connect to make the user assignments.

Classes

SysAdmin:

The SysAdmin class represents the system administrator. The system administrator must have access to everything and be able to see all instances in order to do their job. The SysAdmin is also the person that manages the users in the security facility and must have authority to create views in the DB2 instance where the WDI database resides.

The implementation of the SysAdmin class in 3.2 is as a role with full permissions on every table and no restrictions on any view. There is a file called *sysadmin_perms.ddl* provided with the product that the user can edit as required to create permissions and views for the SysAdmin role. In the file all the GRANT statements contain the full "SELECT, INSERT, UPDATE, DELETE" clause.

EDIAdmin:

The EDIAdmin class represents the EDI administrator. The EDI administrator manages the trading partner community. They must have access to everything necessary to define new trading partners to the system and configure them such that messages flowing to or from the trading partner will be handled correctly. The default set of required permissions are:

Configuration Artifact	Read	Insert	Update	Delete
Mailbox Profile	Y	N	N	N
Network Profile	Y	N	N	N
Network Command Profile	Y	N	N	N
Network Security Profile	Y	N	N	N
MQ Series Queue Profile	Y	N	N	N
Service Profile	Y	N	N	N
MCD Profile	Y	N	N	N
E Envelope Profile	Y	N	N	N
I Envelope Profile	Y	N	N	N
T Envelope Profile	Y	N	N	N
U Envelope Profile	Y	N	N	N
X Envelope Profile	Y	N	N	N
Continuous Receive Profile	Y	N	N	N
Application Defaults Profile	Y	N	N	N
User Exits Profile	Y	N	N	N
CICS Performance Profile	Y	N	N	N
Activity Log Profile	Y	N	N	N
Language Profile	Y	N	N	N
Trading Partner	Y	Y	Y	Y
Contact	Y	Y	Y	Y
EDI Standard Dictionary	Y	N	N	N
EDI Standard Transaction	Y	N	N	N
EDI Standard Segment	Y	N	N	N
EDI Standard Data Element	Y	N	N	N
EDI Standard Code List	Y	N	N	N
EDI Envelope Standard	Y	N	N	N
EDI Envelope Control String	Y	N	N	N
Data Format Dictionary	Y	N	N	N
Data Format Record ID information	Y	N	N	N
Data Format	Y	N	N	N
Data Format Loop	Y	N	N	N
Data Format Record	Y	N	N	N
Data Format Structure	Y	N	N	N
Data Format Field	Y	N	N	N

XML Dictionary	Y	N	N	N
XML DTD	Y	N	N	N
XML Schema	Y	N	N	N
XML Namespace	Y	N	N	N
Data Transformation Map	Y	N	N	N
Rule	Y	Y	Y	Y
Send Map Usage	Y	Y	Y	Y
Receive Map Usage	Y	Y	Y	Y
Print Map report	Y	N	N	N
Transaction Store – All objects	Y	N	N	N

The implementation of the EDIAdmin class in WDI 3.2 is as a role with permissions as shown above and no restrictions on any view.

There is a file called *ediadmin_perms.ddl* provided with the product that the user can edit as required to create permissions and views for the EDIAdmin role.

SystemsIntegrator:

The SystemsIntegrator class represents the role of the person that integrates WDI into the customer's IT systems. The systems integrator manages the queues, networks, files, PERFORM command, service profiles etc. in order to move data between WDI, backend systems and trading partners. They must have access to everything necessary to implement integration of WDI with their own systems. The default set of required permissions are:

Configuration Artifact	Read	Insert	Update	Delete
Mailbox Profile	Y	Y	Y	Y
Network Profile	Y	Y	Y	Y
Network Command Profile	Y	Y	Y	Y
Network Security Profile	Y	Y	Y	Y
MQ Series Queue Profile	Y	Y	Y	Y
Service Profile	Y	Y	Y	Y
MCD Profile	Y	Y	Y	Y
E Envelope Profile	Y	N	N	N
I Envelope Profile	Y	N	N	N
T Envelope Profile	Y	N	N	N
U Envelope Profile	Y	N	N	N
X Envelope Profile	Y	N	N	N
Continuous Receive Profile	Y	Y	Y	Y
Application Defaults Profile	Y	Y	Y	Y
User Exits Profile	Y	Y	Y	Y
CICS Performance Profile	Y	Y	Y	Y
Activity Log Profile	Y	Y	Y	Y
Language Profile	Y	Y	Y	Y
Trading Partner	Y	N	N	N
Contact	Y	N	N	N
EDI Standard Dictionary	Y	N	N	N
EDI Standard Transaction	Y	N	N	N
EDI Standard Segment	Y	N	N	N
EDI Standard Data Element	Y	N	N	N
EDI Standard Code List	Y	N	N	N
EDI Envelope Standard	Y	N	N	N
EDI Envelope Control String	Y	N	N	N
Data Format Dictionary	Y	N	N	N
Data Format Record ID information	Y	N	N	N
Data Format	Y	N	N	N
Data Format Loop	Y	N	N	N
Data Format Record	Y	N	N	N

Data Format Structure	Y	N	N	N
Data Format Field	Y	N	N	N
XML Dictionary	Y	N	N	N
XML DTD	Y	N	N	N
XML Schema	Y	N	N	N
XML Namespace	Y	N	N	N
Data Transformation Map	Y	N	N	N
Rule	Y	N	N	N
Send Map Usage	Y	N	N	N
Receive Map Usage	Y	N	N	N
Print Map report	Y	N	N	N
Transaction Store – All objects	Y	N	N	N

The implementation of the SystemsIntegrator class in 3.2 is as a role with permissions as shown above and no restrictions on any view.

There is a file called *sysintegrator_perms.ddl* provided with the product that the user can edit as required to create permissions and views for the SystemsIntegrator role.

Mapper:

The Mapper class represents the role of the person that manages the transformation maps. They must have access to everything necessary to implement integration of WDI with their own systems. The default set of required permissions are:

Configuration Artifact	Read	Insert	Update	Delete
Mailbox Profile	Y	N	N	N
Network Profile	Y	N	N	N
Network Command Profile	Y	N	N	N
Network Security Profile	Y	N	N	N
MQ Series Queue Profile	Y	N	N	N
Service Profile	Y	N	N	N
MCD Profile	Y	Y	Y	Y
E Envelope Profile	Y	Y	Y	Y
I Envelope Profile	Y	Y	Y	Y
T Envelope Profile	Y	Y	Y	Y
U Envelope Profile	Y	Y	Y	Y
X Envelope Profile	Y	Y	Y	Y
Continuous Receive Profile	Y	N	N	N
Application Defaults Profile	Y	N	N	N
User Exits Profile	Y	N	N	N
CICS Performance Profile	Y	N	N	N
Activity Log Profile	Y	N	N	N
Language Profile	Y	N	N	N
Trading Partner	Y	N	N	N
Contact	Y	N	N	N
EDI Standard Dictionary	Y	Y	Y	Y
EDI Standard Transaction	Y	Y	Y	Y
EDI Standard Segment	Y	Y	Y	Y
EDI Standard Data Element	Y	Y	Y	Y
EDI Standard Code List	Y	Y	Y	Y
EDI Envelope Standard	Y	Y	Y	Y
EDI Envelope Control String	Y	Y	Y	Y
Data Format Dictionary	Y	Y	Y	Y
Data Format Record ID information	Y	Y	Y	Y
Data Format	Y	Y	Y	Y
Data Format Loop	Y	Y	Y	Y
Data Format Record	Y	Y	Y	Y
Data Format Structure	Y	Y	Y	Y
Data Format Field	Y	Y	Y	Y

XML Dictionary	Y	Y	Y	Y
XML DTD	Y	Y	Y	Y
XML Schema	Y	Y	Y	Y
XML Namespace	Y	Y	Y	Y
Data Transformation Map	Y	Y	Y	Y
Rule	Y	Y	Y	Y
Send Map Usage	Y	Y	Y	Y
Receive Map Usage	Y	Y	Y	Y
Print Map report	Y	N	N	N
Transaction Store – All objects	Y	N	N	N

The implementation of the Mapper class in 3.2 is as a role with permissions as shown above and no restrictions on any view.

There is a file called *mapper_perms.ddl* provided with the product that the user can edit as required to create permissions and views for the Mapper role.

Operator:

The Operator class represents the role of the person that manages customer data problems. They must have access to everything necessary to do problem determination and correction of translations involving customer data. The default set of required permissions are:

Configuration Artifact	Read	Insert	Update	Delete
Mailbox Profile	Y	N	N	N
Network Profile	Y	N	N	N
Network Command Profile	Y	N	N	N
Network Security Profile	Y	N	N	N
MQ Series Queue Profile	Y	N	N	N
Service Profile	Y	N	N	N
MCD Profile	Y	N	N	N
E Envelope Profile	Y	N	N	N
I Envelope Profile	Y	N	N	N
T Envelope Profile	Y	N	N	N
U Envelope Profile	Y	N	N	N
X Envelope Profile	Y	N	N	N
Continuous Receive Profile	Y	N	N	N
Application Defaults Profile	Y	N	N	N
User Exits Profile	Y	N	N	N
CICS Performance Profile	Y	N	N	N
Activity Log Profile	Y	N	N	N
Language Profile	Y	N	N	N
Trading Partner	Y	N	N	N
Contact	Y	N	N	N
EDI Standard Dictionary	Y	N	N	N
EDI Standard Transaction	Y	N	N	N
EDI Standard Segment	Y	N	N	N
EDI Standard Data Element	Y	N	N	N
EDI Standard Code List	Y	N	N	N
EDI Envelope Standard	Y	N	N	N
EDI Envelope Control String	Y	N	N	N
Data Format Dictionary	Y	N	N	N
Data Format Record ID information	Y	N	N	N
Data Format	Y	N	N	N
Data Format Loop	Y	N	N	N
Data Format Record	Y	N	N	N
Data Format Structure	Y	N	N	N
Data Format Field	Y	N	N	N

XML Dictionary	Y	N	N	N
XML DTD	Y	N	N	N
XML Schema	Y	N	N	N
XML Namespace	Y	N	N	N
Data Transformation Map	Y	N	N	N
Rule	Y	N	N	N
Send Map Usage	Y	N	N	N
Receive Map Usage	Y	N	N	N
Print Map report	Y	N	N	N
Transaction Store – All objects	Y	N	N	N

The implementation of the Operator class in 3.2 is as a role with permissions as shown above and no restrictions on any view.

There is a file called *operator_perms.ddl* provided with the product that the user can edit as required to create permissions and views for the Operator role.

Last Page