SOLA Administration Guide
Version 6.4.2

WS-Security
Standards-based security enforcement & management
Support for architectural standards
Global dictionary, WSDL first
Management console & dashboard
SLA, monitoring, logging & audit controls
WS-Policy
Standards-based policy enforcement & management
High performance SOAP and REST API engine
Web 2.0 development environment
Change & release management
Version control, backups & migration
Orchestration & microflows
Registry
UDDI, WSRR

Revised: August 2017

Contents

Authorization

Overview

SOLA controls user access in two ways; first, by controlling a user's rights and privileges within SOLA Developer, and second by controlling each user's access to each and every SOLA Container on a project level (different access rights for different projects). In this manner, SOLA can control not only a user's access to the features and controls of SOLA Developer, but also determine what Containers and projects those rights apply to. This combination makes for very powerful and flexible security controls.

A note on terminology: prior versions of SOLA only offered a CICS deployment option, but starting with version 6.0, SOLA had the ability to integrate with IMS from a CICS container. With the release of the SOLA IMS plug-in, we have removed the CICS dependency and are now shipping a Started Task Address Space version of the SOLA runtime (managed through Resource Manager). For this reason, we will no longer refer to the SOLA runtime a "SOLA TOR." It will henceforth be referred to as a "SOLA Container."

SOLA recognizes three levels of user access to SOLA Developer; administrator, programmer and public access. Furthermore, there are two categories of administrators with escalating privileges (detailed below).

The three levels of user access are as follows:

- **Public Access:** this is the lowest access level and can do the following:
  - Directory search
  - View WSDL
  - Quick Test function pointing to test containers

- **Programmer:** this type of user can access all of SOLA's functional features, but lacks access to security and administration options. This user can import and analyze programs and resources in projects in which the user is authorized to work.

- **Administrator:** consists of two categories, whose escalating privileges are detailed below.

SOLA recognizes two levels of administration, with each higher level encompassing all of the functions of the levels below. These are:

- **Project Administration**
- **SOLA Administration**

A **Project Administrator** has the following access rights defined and assigned by the SOLA Administrator:

- Inquire on and update policy associated with a project, program, method or operation in any run-time environment (includes auditing).
- Delete a project/program/method.
- Add a new user to the project.
- Check environment setup.
- Check Monitor and/or Error Logs.
- Read/Disable/Enable a template.
- Is automatically assigned limited access to the SOLA Administration File and Property Editor (see page 41).

A **SOLA Administrator** has all of the access rights of the Project Administrator, as well as the following additional access rights:

- Change installation defaults.
- Pre-load a certificate to SOLA.
- Check web-site user activity log.
- Register/maintain Listeners.
- Add/Remove another SOLA Administrator.
- Add/Remove a Project Administrator.
- Update SOLA system files.
- Update SOLA web-site properties.
- Update User properties.
- SOLA Administrators do not belong to access control groups as they have full access to all SOLA containers.

To perform any of the functions above, the user must log in.

In addition to setting an account type (programmer, administrator) for each SOLA user, the user's access to containers can be controlled by creating container groups and specifying which groups the user has access to within each project.

For example, let's say your environment consists of five containers; two test containers, a QA container and two production containers. Using SOLA's security controls, you can group your containers into one or more container groups. For this example, let’s assume you decide to create three container groups and call them TEST, STAG(stage) and PROD. You can then assign several TEST TOR Region CICS SYSID’s (test containers) to the TEST group, your STAG TOR Region CICS SYSID (stage container) to your STAG group, and perhaps several Production TOR Region CICS SYSID’s (Production containers) to your PROD group.

It is therefore possible to create a user account that has access to all SOLA container groups in a TOR region(s), or access to one container group in a single TOR region.

User Authorization

With the introduction of the SOLA Resource Manager, there are now three methods of managing SOLA Users; the User Authority Manager panel when right clicking on a program in the Directory tree in SOLA Developer, the SOLA Admin Menu in SOLA Developer and the Users tab in Resource Manager.

A user that creates a project is automatically designated a Project Administrator for that project. Additional Project administrators can be added using the User Authority dropdown Menu associated with the project by right clicking on the Project (discussed later in this section). A Project Administrator has access to project and method-level administration features.

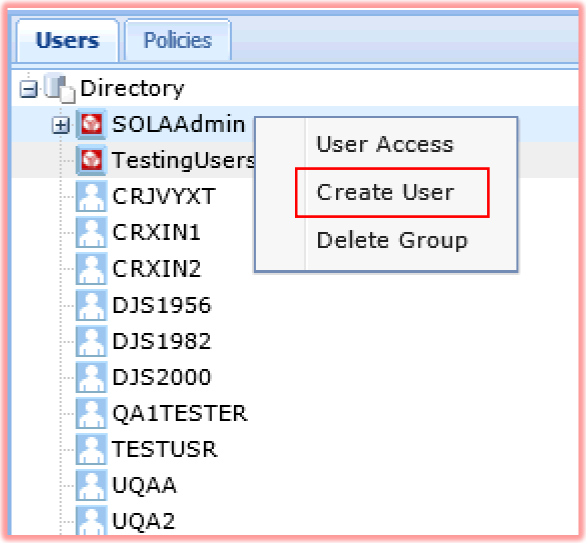
Creating and Managing Users

To create a new user using Resource Manager, right click the directory root and select **Create User** from the menu.

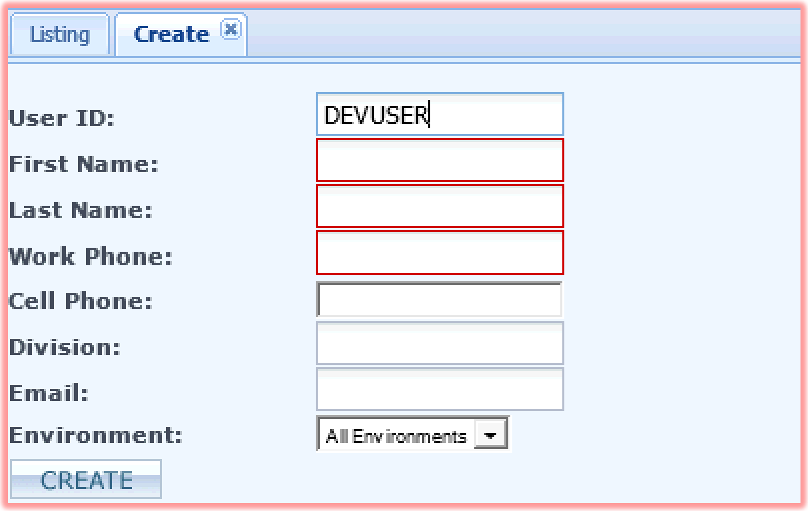
This will display the Create tab and allow you to create a new user account by filling in the user account name and contact information for the user. Fields outlined in red are required.

To create a user account in a specific group (such as SOLAAdmin), right click the group name instead of the directory root and select **Create User** from the group menu.

Creating a new user in Resource Manager is identical to creating a user account in SOLA Developer, with one minor difference; the access level of the account (either programmer or administrator) depends on whether or not the user is in the **SOLAAdmin** group (see below).



User accounts created in Resource Manager will be available for use in SOLA Developer, and users created in SOLA Developer will likewise be available in Resource Manager and will appear in the users tab in the subjects panel.



User ID:	DEVUSER
First Name:	
Last Name:	
Work Phone:	
Cell Phone:	
Division:	
Email:	
Environment:	All Environments

Add/Update User Authorization within SOLA Developer

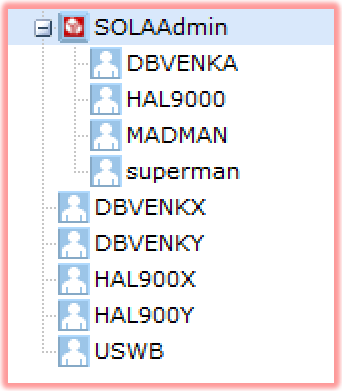
To add or modify user authorization for a specific project, right click on the project name in SOLA Developer and select **User Authority**. Doing so will display the **User Authority Manager** panel, which allows you to authorize a user to work on the project. Several levels of authorization are available. Each level of authorization grants all access rights of the previous levels.

Note: The SOLA Developer Users Guide will explain User Authorization in detail.

The following access levels are available:

- **Project Admin:** grants full access to the project, which includes the ability to delete the project and add or remove users.
- **Programmer:** grants full access to the project.
- **Import:** allows the user to import programs.
- **Analyze**: allows the user to create methods.
- **QuickTest:** allows the user to test methods using the SOLA test harness.
- **Promote:**
- **Demote:**
- **Recover:**
- **Insert:**
- **Update:**
- **Delete:** allows the user to delete programs in the project.

User Groups



SOLAAdmin
- DBVENKA
- HAL9000
- MADMAN
- superman

DBVENKX
DBVENKY
HAL900X
HAL900Y
USWB

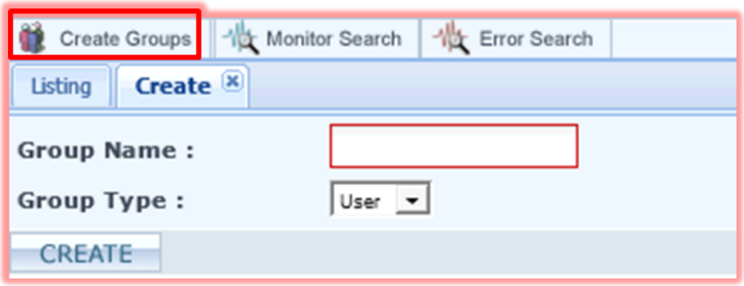
User groups serve two purposes; organizing users to create multi-user accesses and determining which users have administrator access.

For the latter purpose, Resource Manager comes with a default group called “SOLAAdmin”. Users created in that group, or dragged into that group, are SOLA Administrators with full administrative access rights to SOLA Developer.

If you delete the SOLAAdmin group, you will delete all users in that group and delete all SOLA Administrator accounts. If the group has been deleted, you can create a new group called “SOLAAdmin”, and that group will function exactly as the original system group.

Other than the SOLAAdmin group, groups serve no purpose other than to organize user accounts and make creating user accesses easier. Instead of dragging individual users onto a resource to create an access, you can drag a user group to create accesses for all the users in that group.

You can drag a user from the root directory or another group into a new user group, as well as drag a user out of a group and into the root directory.



Group Name :

Group Type : User

To create a new user group, click the Create Groups button. This will display the Create tab and allow you to create a group.

Select “User” from the **Group Type** menu, then enter the group name.

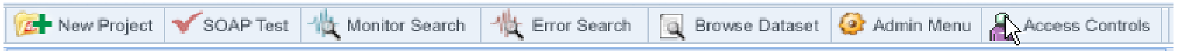
When you are ready to create the group, click **Create**.

Alternate Ids - Granting Mainframe FTP Access

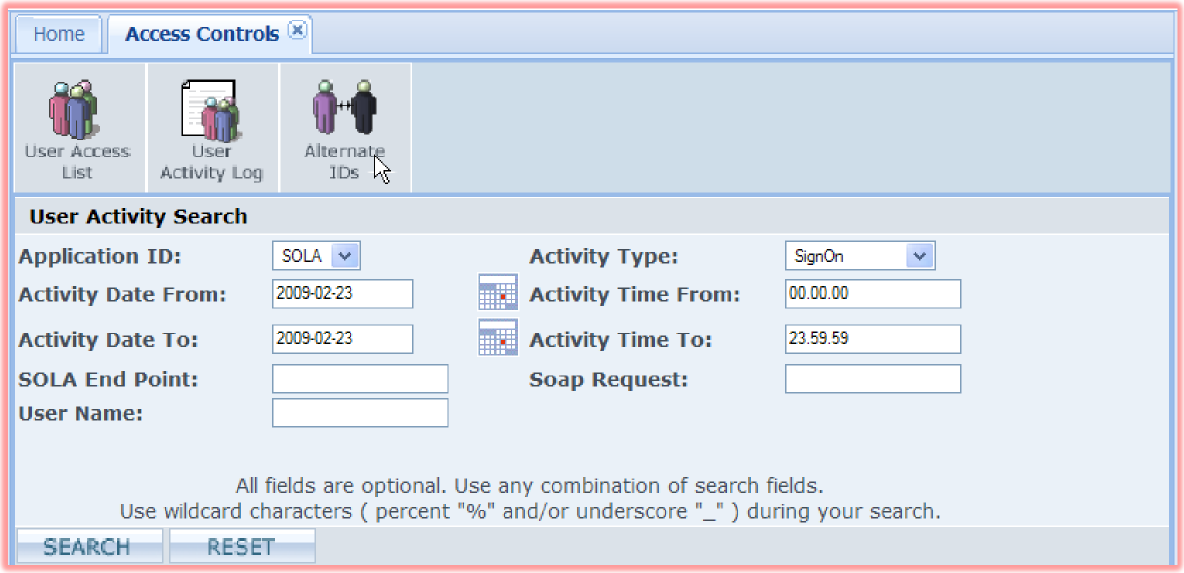
For those SOLA users that do not have mainframe ftp access, a SOLA Administrator can grant such access (through SOLA only) by defining an alternate (generic) user Id. Alternate Ids are SOLA user Ids that have mainframe FTP access. This can be an existing user's Id or an Id created specifically for the purpose of granting FTP access. There is no limit to how many alternate/generic user Ids can be defined.

Once an alternate Id is defined, a Project Administrator can assign the Id to an authorized project user who does not otherwise have mainframe ftp access. Doing so will grant that user the ability to use FTP functions in SOLA, such as browsing a dataset, importing a program, or finalizing an analysis.

To define alternate user Ids, access the SOLA Access Controls screen by selecting **Access Controls** from the SOLA Developer Menu Bar.



The following screen will be displayed:

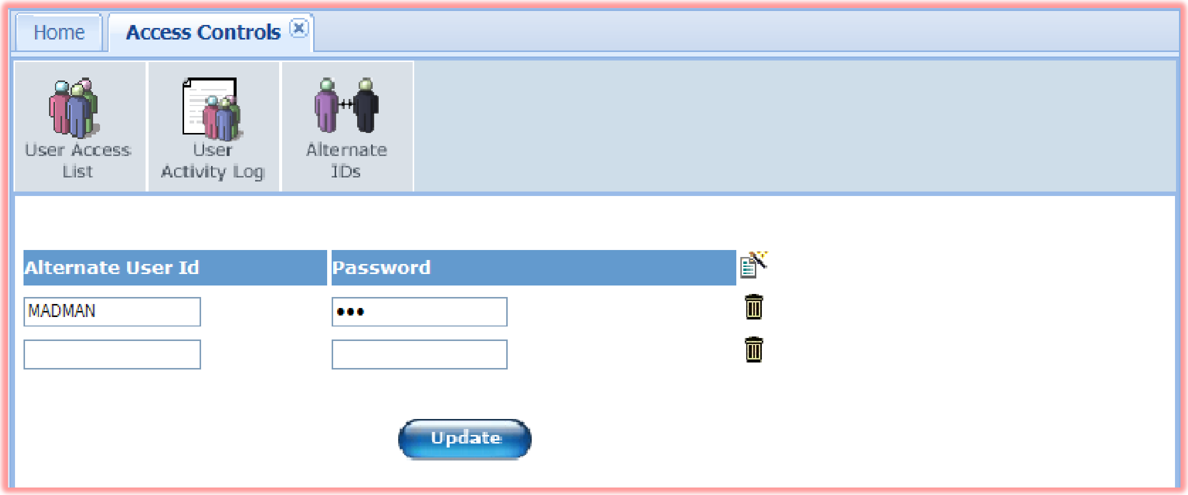


User Activity Search

Application ID:	SOLA	**Activity Type:**	SignOn
Activity Date From:	2009-02-23	**Activity Time From:**	00.00.00
Activity Date To:	2009-02-23	**Activity Time To:**	23.59.59
SOLA End Point:		**Soap Request:**	
User Name:			

All fields are optional. Use any combination of search fields.
Use wildcard characters (percent "%" and/or underscore "_") during your search.

Click on the Alternate IDs button to access the Alternate IDs screen.



Alternate User Id	Password
MADMAN	•••

Enter the alternate ID and password that can be used to execute FTP to the mainframe.

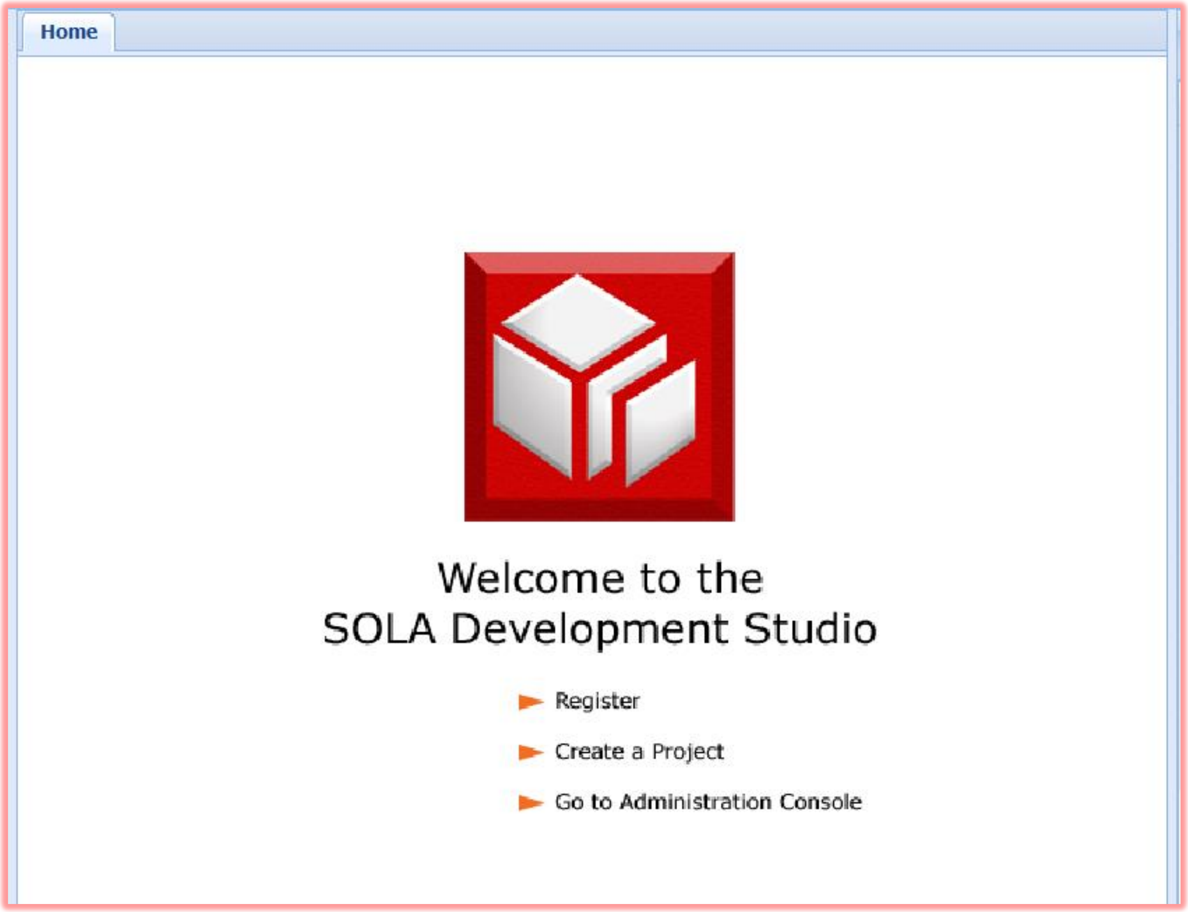
To add additional rows, click the 📝 button. To remove an unwanted row, click the 🗑 button.

When you are finished, click **Update**.

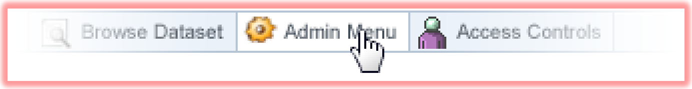
System Properties

Customization of the SOLA Home Page

SOLA ships with a default home page, as shown below. This page can be customized to your installation's requirements.

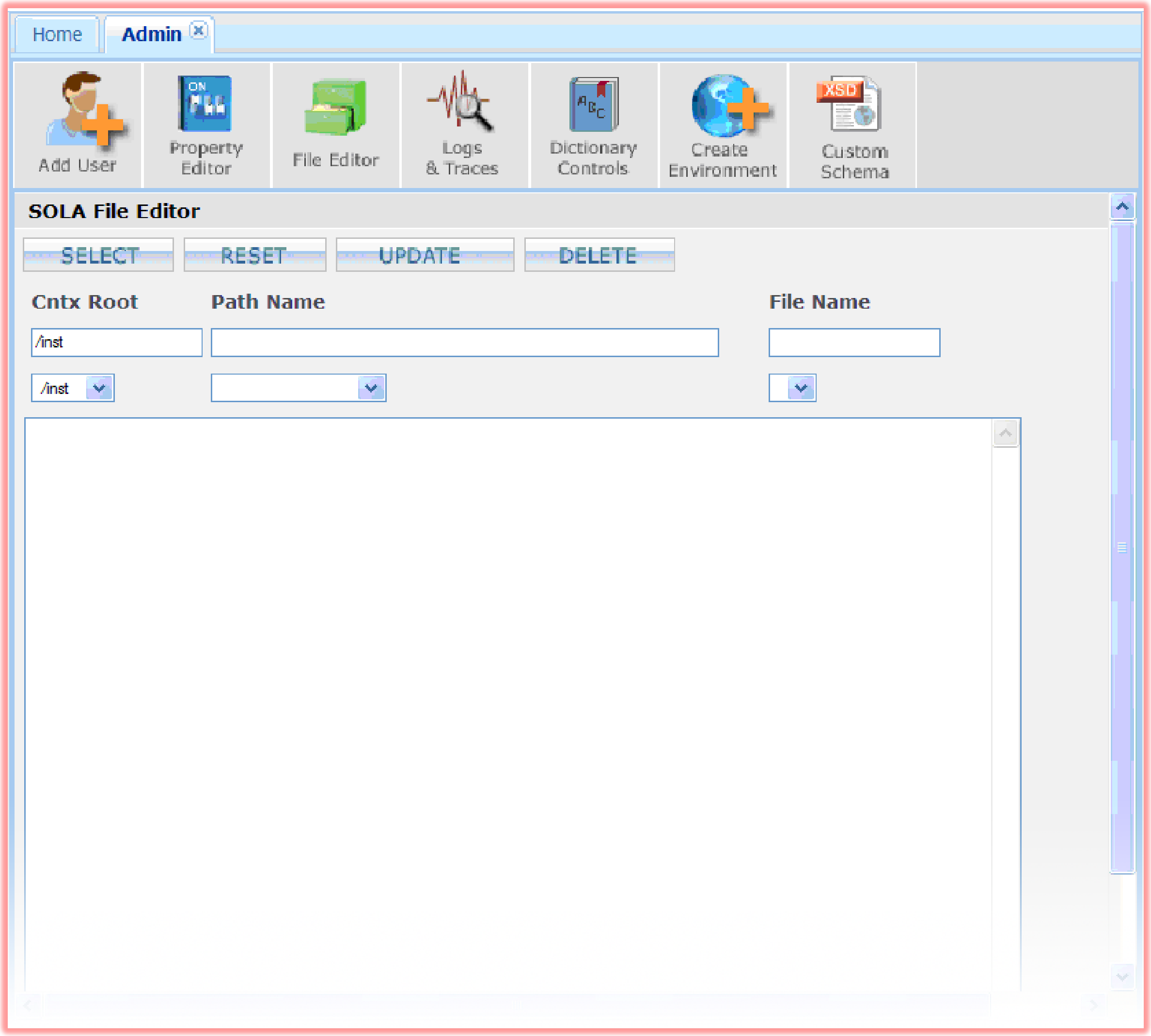


To access SOLA Developer's administration functions, click on the **Admin Menu** button in the button bar.



This will display the admin console.

Click on the **File Editor** icon to display the Property / File Editor screen set to File Editor mode.



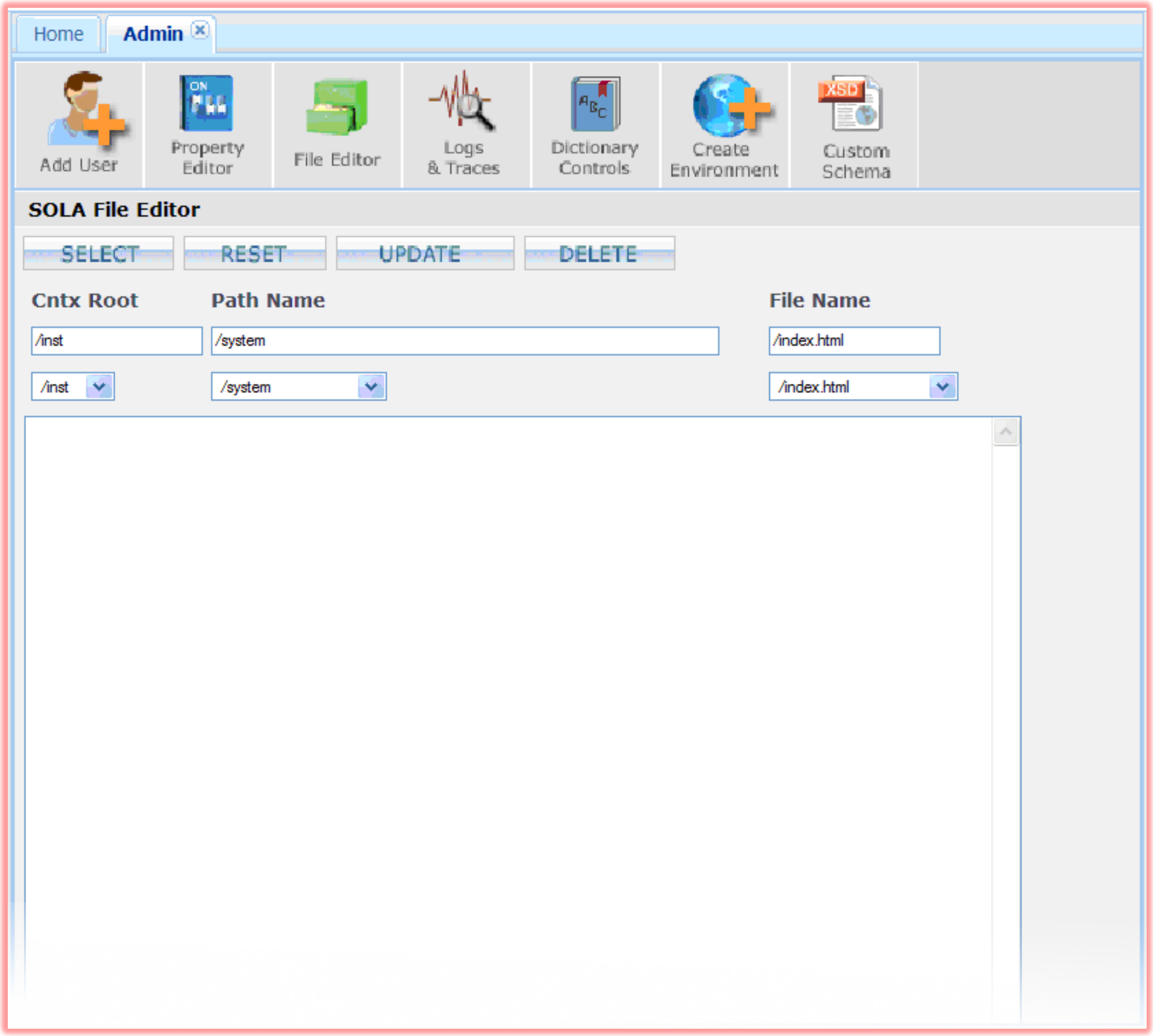
Cntx Root: /inst

Path Name:

File Name:

Cntx Root dropdown: /inst

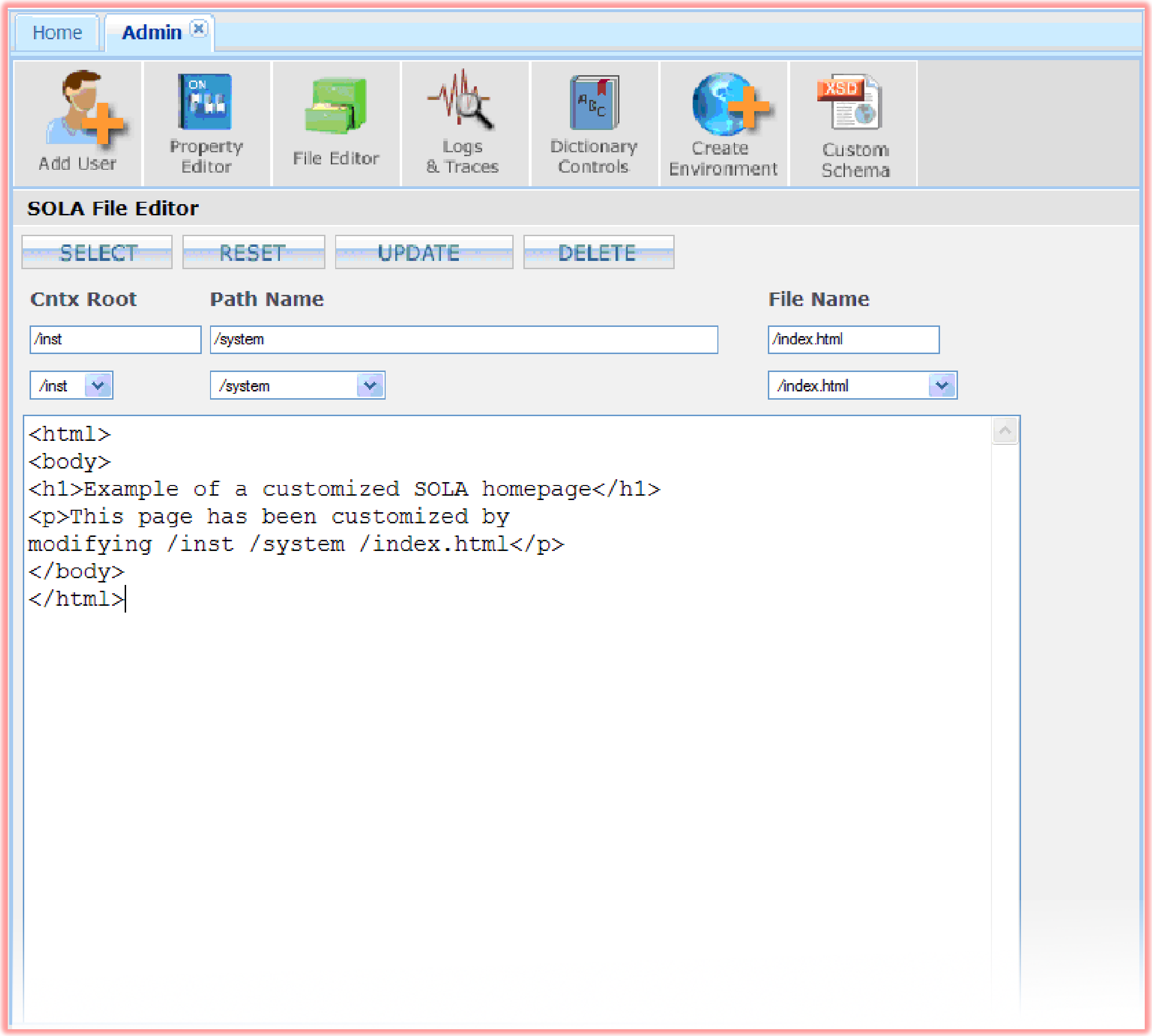
Choose /inst /system /index.html from the three dropdown boxes, and click the **SELECT** button.



SOLA File Editor

Cntx Root	Path Name	File Name
/inst	/system	/index.html
/inst	/system	/index.html

Enter a valid HTML page and then press the **UPDATE** button.



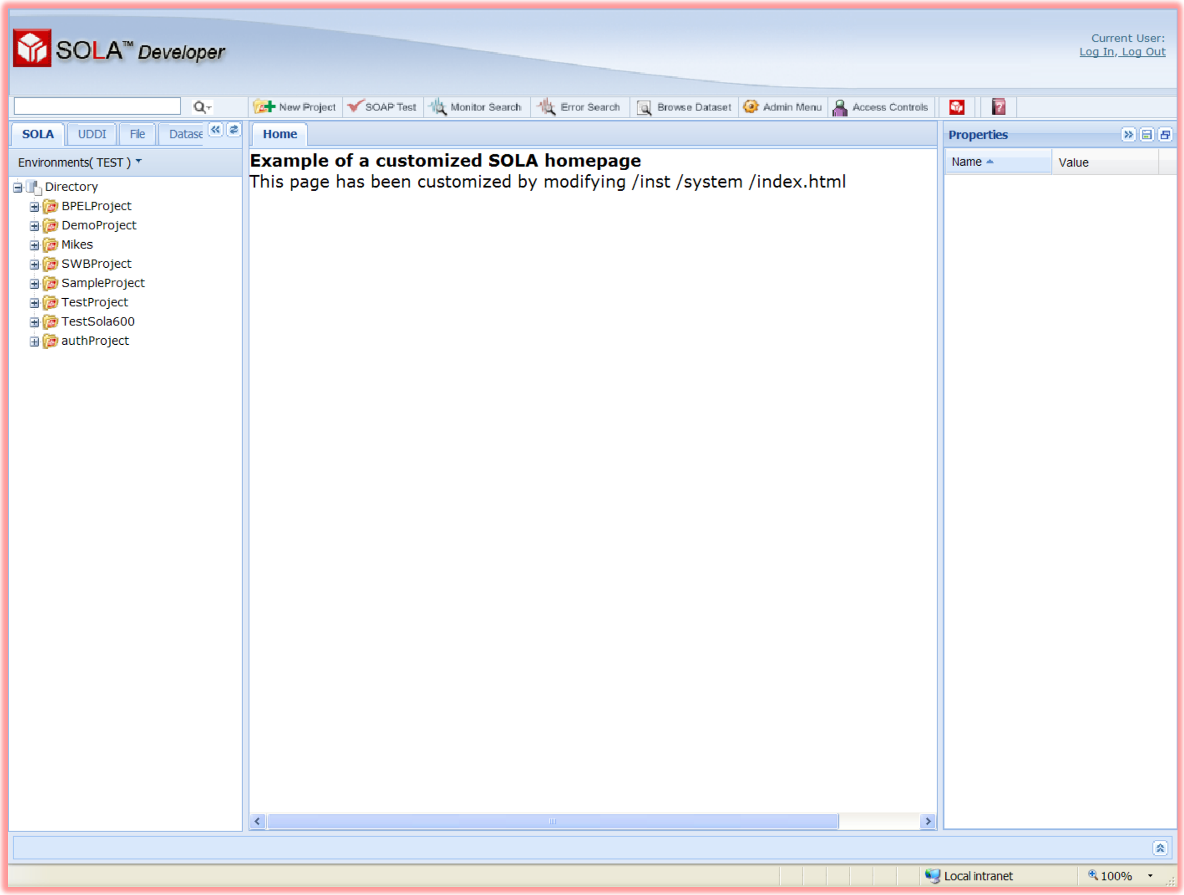
SOLA File Editor

Cntx Root: /inst
Path Name: /system
File Name: /index.html

/inst
/system
/index.html

```
<html>
<body>
<h1>Example of a customized SOLA homepage</h1>
<p>This page has been customized by
modifying /inst /system /index.html</p>
</body>
</html>
```

The following is an example of the SOLA index page with the sample changes.



Example of a customized SOLA homepage

This page has been customized by modifying /inst /system /index.html

Installation Default Changes

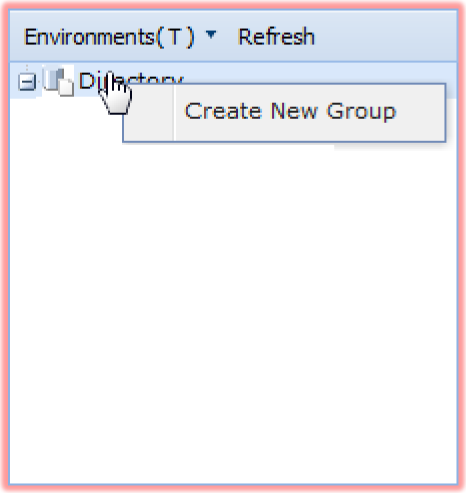
SOLA installation defaults are set at the Container Group level. You can display and/or modify the defaults for a Container Group by using the SOLA Resource Manager.

Container Groups

Container groups are more than just containers for groups; they also serve to control metrics collection and security policies for the containers inside them. Groups allow you to do the following:

- Enable and configure metrics collection
- Enable use of the default security policy
- Designate a security user exit
- Configure cache, queue and storage options
- Enable and configure custom security policy

Creating Container Groups

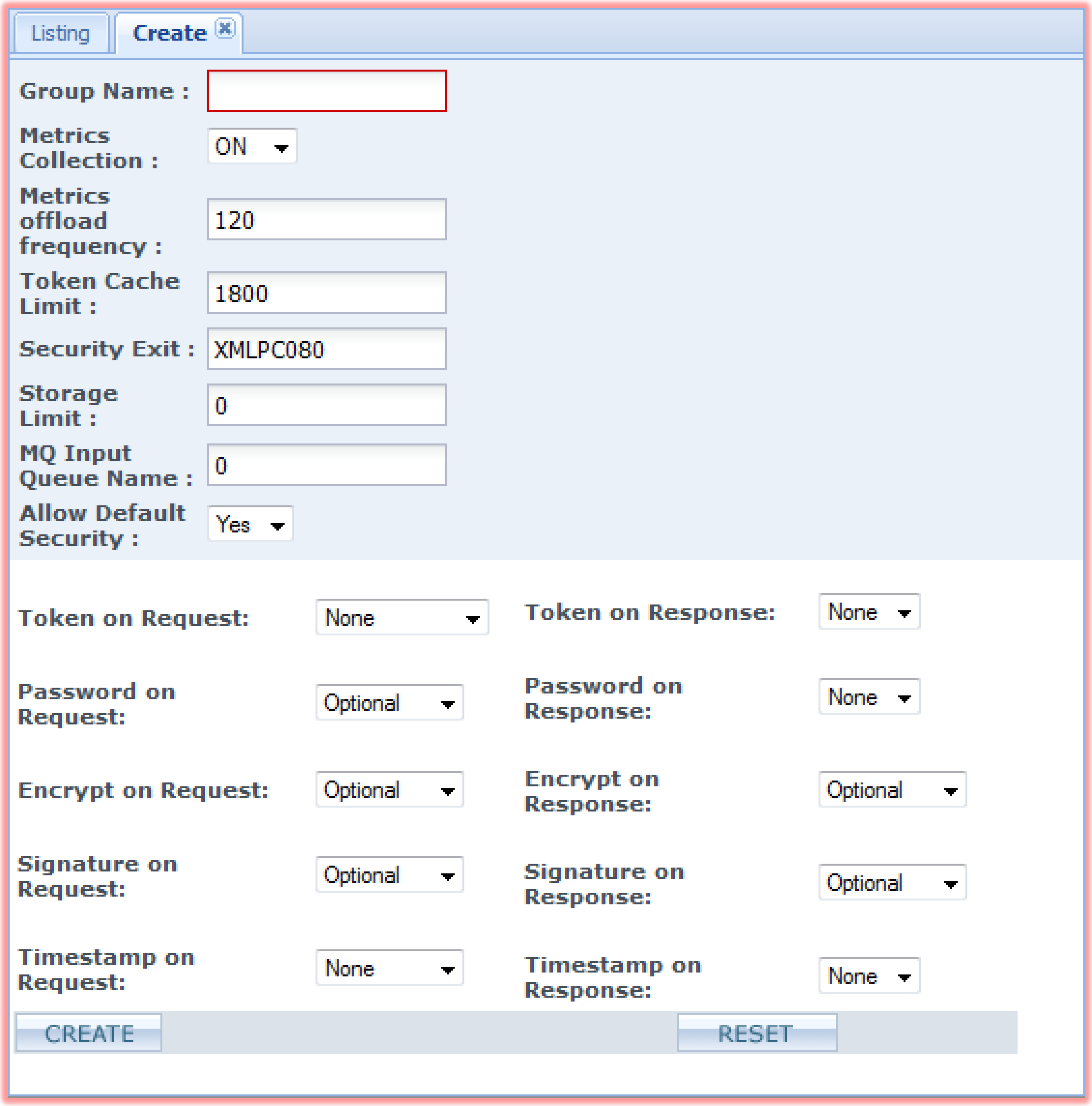


All containers in Resource Manager must be contained in a container group. To create a container group, right click on the directory icon and select **Create New Group** from the menu.

This will display the Create tab in the workspace, allowing you to create a new container group.

We recommended that you group your containers based on their security level (low security group for test containers, high security group for production containers, etc.). This will make assigning access a lot simpler.

The create tab contains a series of fields that you will need to populate to create a new group. All fields/menus except for the group name are preconfigured with the default settings.



Group Name :

Metrics Collection : ON

Metrics offload frequency : 120

Token Cache Limit : 1800

Security Exit : XMLPC080

Storage Limit : 0

MQ Input Queue Name : 0

Allow Default Security : Yes

Token on Request:	None	Token on Response:	None
Password on Request:	Optional	Password on Response:	None
Encrypt on Request:	Optional	Encrypt on Response:	Optional
Signature on Request:	Optional	Signature on Response:	Optional
Timestamp on Request:	None	Timestamp on Response:	None

To create a group with the default settings, fill in the **Group Name** field and click **CREATE**. To configure custom settings for the group, you will need to make changes to the following settings.

Standard Settings:

- **Metrics Collection:** enables (ON) or disables (OFF) metrics collection (by SOLA) for the containers in the group.

- **Metrics offload frequency:** determines how often, in seconds, metrics are spooled to the database.

- **Token Cache Limit:** How long, in seconds, before a cached token expires. The lowest limit is 10 seconds and anything below 10 will reset it to 1200 seconds. After updating this value, wait 3 minutes for it to take effect.

- **Security Exit:** specifies the program to be used as security exit. By default, XMLPC080, the SOLA security exit, is used

- **Storage Limit:** the maximum size of an outbound message

- **MQ Input Queue Name:** the name of the MQ queue that SOLA will listen to for input MQ messages

- **Allow Default Security:** specifies whether the containers in the group will use the default security policy. Choosing "No" will force the containers in the group to use the custom security policy, defined below

Security Policy Settings: These settings create a default policy for the Container Group.

- **Token on Request:** this setting determines whether SOLA will accept requests without an attached security token.
  - **NO:** SOLA will allow requests without security tokens
  - **MainframeID:** SOLA will require a mainframe user id as a security token.
  - **LDAP ID:** SOLA will require an LDAP user id as a security token.
  - **SAML:** SOLA will require SAML credentials as a security token.
  - **Restrict by IP:** whether only certain IP addresses can submit requests

- **Token on Response:** with the current version of SOLA, the only option is NO.

- **Password on Request:** this setting determines whether SOLA accepts requests that have a token, but no password
  - **Optional:** a password is not required (SOLA will accept requests without a password).
  - **Mandatory:** a password is required.

- **Password on Response:** with the current version of SOLA, the only option is NO.

- **Encrypt on Request:** this setting determines whether SOLA accepts requests that are not encrypted.
  - **Optional:** encryption is not required (SOLA will accept requests without encryption).

    - **Mandatory:** encryption is required.

- **Encrypt on Response:** this setting determines whether SOLA will encrypt responses
    - **Optional:** SOLA will not encrypt responses
    - **Mandatory:** SOLA will encrypt responses

- **Signature on Request:** this setting determines whether SOLA accepts requests without an attached signature.
    - **Optional:** attached signatures are not required (SOLA will accept requests without attached signatures).
    - **Mandatory:** the body of the SOAP request must be signed.

- **Signature on Response:** this setting determines whether SOLA will attach a signature to responses.
    - **Optional:** SOLA will not attach a signature to responses.
    - **Mandatory:** SOLA will attach a signature to responses.

- **Timestamp on Request:** this setting determines whether SOLA accepts requests without an attached timestamp. The timestamp contains the policy’s expiration date and time.
    - **None:** attached timestamps are not required (SOLA will accept requests without attached timestamps).
    - **Mandatory:** attached timestamps are required.

- **Timestamp on Response:** with the current version of SOLA, the only option is NO.

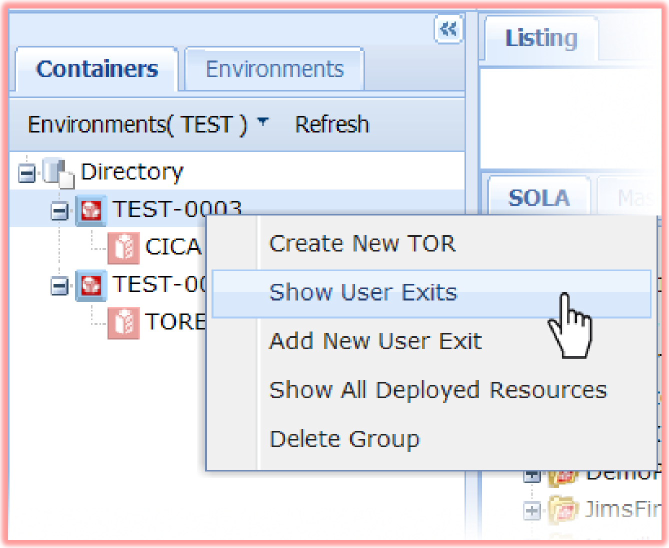
When you are finished configuring the group, click **CREATE**. You can reset all the settings to their defaults at any time clicking the **RESET** button.

Configuring User Exits

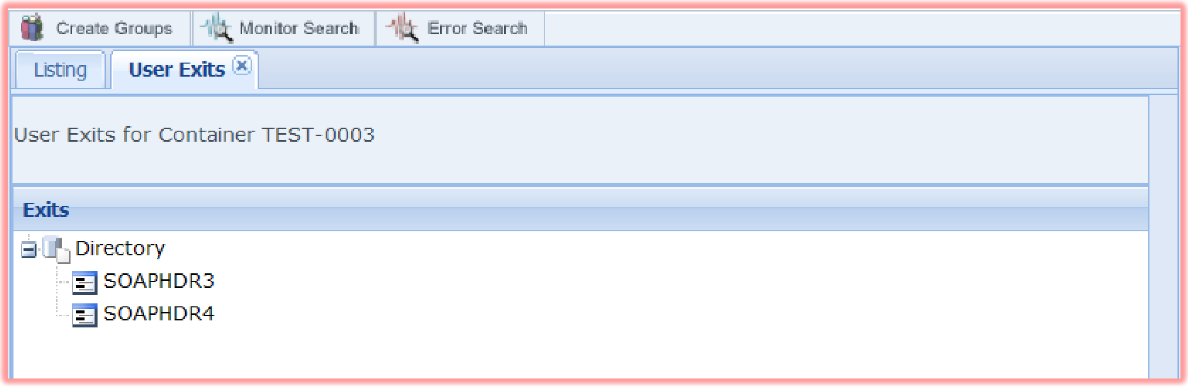
User exits are managed using the SOLA Resource Manager. SOLA allows you to run user exits at three points in the processing of a SOAP message:

1. Processing the input SOAP Header
2. Processing the output SOAP Header
3. Modifying the statistics written by the SOLA logger

To configure user exit settings, right click on the Container Group that you'd like to manage in the SOLA Containers tree.

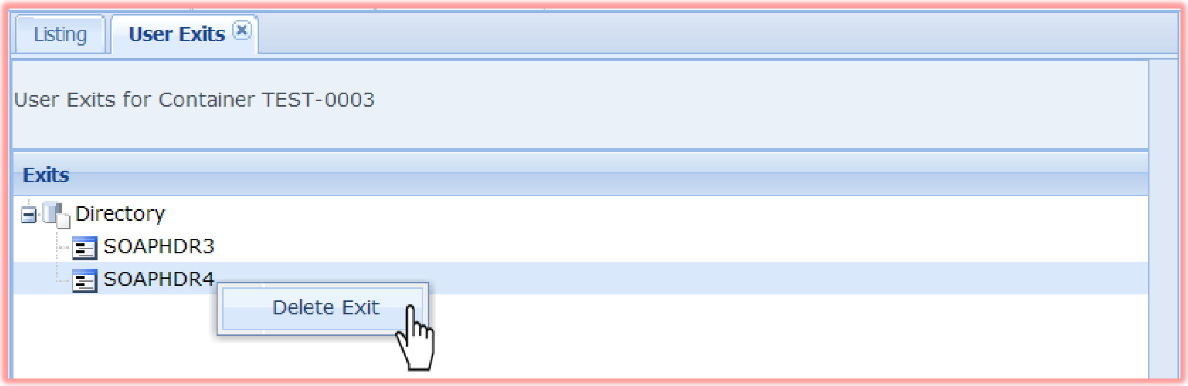


The following screen will be displayed:



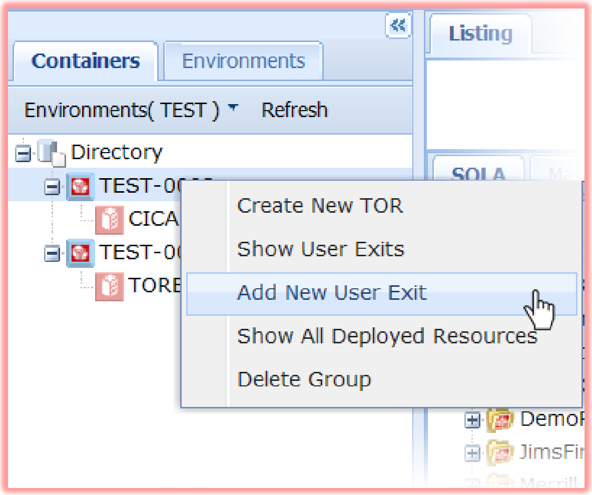
User Exits for Container TEST-0003

This shows two user exits, SOAPHDR3and SOAPHDR4. The only function that can be performed from this screen is to delete a user exit. Right click on the exit you'd like to delete.

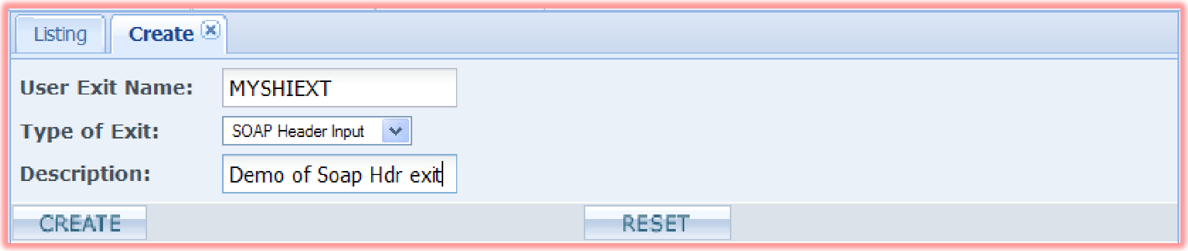


User Exits for Container TEST-0003

To add a user exit, right click on the Container Group that you'd like to add the exit to:



This will display the following screen:



User Exit Name: MYSHIEXT

Type of Exit: SOAP Header Input

Description: Demo of Soap Hdr exit

User Exit Name: this is the name of the user exit program that will be called by SOLA. If you're running in a SOLA CICS Container then this program must have appropriate CICS table entries.

Type: this menu is used to specify exit type, which determines when the exit is called. Currently, there are three options:

- **Soap Header Input**. This exit is called before the execution of the target legacy program.
- **Soap Header Output**. This exit is called before or after the target legacy program has been executed and the ‘Envelope’ tag for the outbound SOAP response has been created.
- **Status**. (SOAP Response). This exit is called under one of two circumstances; either when the body of out an outbound SOAP response has been built by SOLA or when SOLA is about to throw a SOAP fault.

Description: this output field contains a description of the exit type.

Writing a User Exit

Overview

SOLA user exits were designed to provide access, at various points, to the SOLA SOAP stack. The exits currently supported are the **SOAP Header In**, **SOAP Header Out**, and the **User Status Exit**.

Depending on the exit, SOLA's behavior can be altered in a variety of ways, some of which are detailed below.

- Stop processing and throw a SOAP fault as defined by the user exit.
- Stop processing and return a custom SOAP response as defined by the user exit.
- Alter the contents of a SOAP message, either inbound or outbound. Possibilities include adding SOAP header information or altering the contents of a SOAP request or response.
- Stop execution of the target legacy program.
- Perform any installation specific processing such as maintaining monitoring or status data in a data store other than SOLA's default monitoring tables.

User Exit Description

SOAP Header Processing Inbound

The Soap Header Processing Inbound exit is called before the execution of the target legacy program. At this point, the user exit program can interrogate (using the SOLA DOM parser) the contents of the SOAP header or any part of the SOAP request.

Based on the user exit's logic, it can instruct SOLA to discontinue processing and throw a fault or generate a custom response. It can also inform SOLA of the region to which the legacy program request should be directed or just store information for later use.

SOAP Header Processing Outbound

The SOAP Header Processing Outbound exit is called before or after the target legacy program has been executed and the 'Envelope' tag for the outbound SOAP response has been created.

At this point, the user exit program can append a custom SOAP header to the outbound SOAP response or instruct SOLA to stop processing and throw a custom fault or a custom SOAP response.

Outbound Status

The Outbound Status exit can be called under one of two circumstances.

After the ‘Body’ of the outbound SOAP response has been built by SOLA, this user exit is called and given the opportunity to add status data to the outbound request, record monitoring statistics or instruct SOLA to throw a custom fault or custom SOAP response.

This exit can also be called when SOLA is about to throw a soap fault. In this case you can instruct SOLA to continue with its own fault processing, instruct SOLA to send a custom SOAP fault, or append data to the soap response and deliver the altered response instead of throwing a soap fault.

User Exit API

All user exit types share a common User Exit API. This data structure is passed to each user exit as a COMMAREA and is included in the SOLA package under samplib as a copy book called SOLAEXIT.

When using this copy book, the ‘:WS-:’ string needs to be replaced with a string of your choosing.

For Example:

```
01  DFHCOMMAREA.
    COPY SOLAEXIT REPLACING ==:WS-:== BY ==LK-==.
```

Copy Book SOLAEXIT

The following copybook can be found in your SAMPLIB under the name EXITCOPY.

```
     05  :WS-:Return-Code                PIC S9(04) BINARY.
         88  :WS-:Normal-Continue        VALUE +0.
         88  :WS-:Request-Authorized     VALUE +0.
         88  :WS-:Throw-Sola-Fault       VALUE -1.
         88  :WS-:Throw-Custom-Fault     VALUE -2.
         88  :WS-:Custom-Response        VALUE -3.
         88  :WS-:REQUEST-REJECTED       VALUE -4 -10 -11.
         88  :WS-:Credentials-Expired    VALUE -10.
         88  :WS-:Invalid-Userid-PSW     VALUE -11.
         88  :WS-:PROGRAM-ERROR          VALUE -5.
         88  :WS-:DB2-ERROR              VALUE -6.
         88  :WS-:PROG-ABEND             VALUE -9999.
     05  :WS-:Sola-Provided-Data.
         10  :WS-:Sola-Exit-Type         PIC  X(03).
             88  :WS-:Soap-Header-In     VALUE 'SHI'.
             88  :WS-:Soap-Header-Out    VALUE 'SHO'.
             88  :WS-:Status-Msg         VALUE 'STA'.
             88  :WS-:Status-Log         VALUE 'LOG'.
         10  :WS-:Sola-Status            PIC S9(04) BINARY.
             88  :WS-:Status-Success     VALUE +0.
             88  :WS-:Status-Failure     VALUE -1.
         10  :WS-:Sola-Error-Type        PIC S9(04) BINARY.
             88  :WS-:No-Error           VALUE +0.
             88  :WS-:Application-Error  VALUE -1.
             88  :WS-:System-Error       VALUE -99.
         10  :WS-:Sola-Status-Msg        PIC  X(254).
         10  :WS-:Client-Ip-Addr         PIC  X(15).
         10  :WS-:Cics-Sysid-Tor         PIC  X(04).
         10  :WS-:Cics-Sysid-Aor         PIC  X(04).
         10  :WS-:Task-Start-Time-X.
             15  :WS-:Task-Start-Time PIC S9(15) VALUE +0 COMP-3.
         10  :WS-:Legacy-Prog-Nm         PIC  X(08).
```

```
    10  :WS-:Legacy-Prog-Typ        PIC  X(02).
        88  :WS-:Commarea-Map       VALUE 'CA'.
        88  :WS-:Callable           VALUE 'CL'.
        88  :WS-:Custom             VALUE 'CU'.
        88  :WS-:BMS3270            VALUE 'BM'.
        88  :WS-:AdhocSQL           VALUE 'SQ'.
        88  :WS-:DB2SP              VALUE 'SP'.
        88  :WS-:VSAM-PLUGIN        VALUE 'VS'.
    10  :WS-:Template-Nm            PIC  X(08).
    10  :WS-:Method-Nm              PIC  X(35).
    10  :WS-:Method-Ns              PIC  X(256).
    10  :WS-:Parent-Tag-Nm          PIC  X(50).
    10  :WS-:User-Token             PIC  X(128).
    10  :WS-:User-Creds          REDEFINES :WS-:User-Token.
        15  :WS-:User-Id         PIC  X(20).
        15  :WS-:User-Psw        PIC  X(20).
        15  FILLER               PIC  X(88).
    10  :WS-:Dom-Ptr                USAGE IS POINTER.
    10  :WS-:Soap-Header-Ptr        USAGE IS POINTER.
    10  :WS-:Soap-Header-Len        PIC S9(09) BINARY.
    10  :WS-:Soap-Req-Ptr           USAGE IS POINTER.
    10  :WS-:Soap-Req-Addr REDEFINES :WS-:Soap-Req-Ptr
                                    PIC S9(09) BINARY.
    10  :WS-:Soap-Req-Len           PIC S9(09) BINARY.
    10  :WS-:Soap-Resp-Ptr          USAGE IS POINTER.
    10  :WS-:Soap-Resp-Len          PIC S9(09) BINARY.
    10  :WS-:Commarea-Area.
         15  :WS-:Commarea-Ptr      USAGE IS POINTER.
         15  :WS-:Commarea-Len      PIC S9(09) BINARY.
    10  :WS-:Bms-Map-Area REDEFINES :WS-:Commarea-Area.
         15  :WS-:Map-Ptr           USAGE IS POINTER.
         15  :WS-:Map-Len           PIC S9(09) BINARY.
    10  :WS-:PSW-START-POS          PIC S9(09) VALUE +0 BINARY.
    10  :WS-:PSW-LEN                PIC S9(04) VALUE +0 BINARY.
    10  :WS-:POLICY-DATA            PIC  X(494) VALUE SPACES.
    10  :WS-:Filler                 REDEFINES :WS-:POLICY-DATA
                                    PIC  X(494).
05  :WS-:Exit-Response-Data.
    10  :WS-:Logging-Ind            PIC  X(01).
        88  :WS-:Log-Internal-Msg   VALUE 'Y'.
    10  :WS-:Error-Code             PIC S9(04) BINARY.
    10  :WS-:Fault-String           PIC  X(40).
    10  :WS-:Internal-Msg           PIC  X(100).
    10  :WS-:External-Msg           PIC  X(40).
    10  :WS-:Custom-Resp-Ptr        USAGE IS POINTER.
    10  :WS-:Custom-Resp-Len        PIC S9(09) BINARY.
    10  :WS-:Legacy-Sysid           PIC  X(04).
    10  :WS-:Fault-Code             PIC  X(01).
        88  :WS-:Server             VALUE 'S'.
        88  :WS-:Client             VALUE 'C'.
    10  :WS-:ABEND-CD               PIC  X(04) VALUE LOW-VALUES.
    10  :WS-:SQLCODE                REDEFINES
        :WS-:ABEND-CD               PIC S9(09) BINARY.
    10  :WS-:ACTION-ON-ERROR        PIC  X(01) VALUE 'R'.
        88  REJECT-ON-ERROR         VALUE 'R'.
        88  ALLOW-ON-ERROR          VALUE 'A'.
    10  :WS-:Legacy-Tranid          PIC  X(04).
05  :WS-:Scratch-Area               PIC  X(491).
```

Copybook Variables

There are two types of variables in the copybook; input variables and output variables. Input variables are passed to the User Exit by SOLA. Output variables are passed to SOLA by the User Exit program.

Input Variables

Sola-Exit-Type

SOLA uses this field to notify the user exit program of the exit type being invoked. This is useful if the same program is being used for more than one exit type.

Sola-Status

Notifies the user exit of SOLA’s status at the time the exit was called. A value of 0 (zero) indicates that SOLA processing is normal at the time of the exit’s invocation. A value of -1 indicates that SOLA has called the user exit during fault processing.

Sola-Error-Type

When the value of the Sola-Status variable is -1, this variable indicates to the user exit whether the error encountered by SOLA was a SOLA system error or a legacy application error.

Sola-Status-Msg

When the value of the Sola-Status variable is -1, this variable will contain the message that SOLA intends to use as “fault text” when SOLA throws a SOAP fault.

Client-Ip-Addr

The IP address (if available) of the client that originated the request. This is usually an application server.

Cics-Sysid-Tor

The four character CICS SYSID for the SOLA TOR region.

Cics-Sysid-Aor

The four character CICS SYSID for the legacy AOR region (if available).

Task-Start-Time
The time at which the task was started.

Legacy-Prog-Nm
The name of the legacy program that is targeted by the web service.

Legacy-Prog-Typ
The legacy program type (e.g. ‘CA’ = Commarea). For more information, see the 88 level descriptions in the copybook.

Legacy-Tranid
Instructs SOLA to DPL the legacy program using the transaction Id indicated in this field. Make sure that this transaction id is defined as a mirror transaction (pointing to DFHMIRS) in the AOR.

Template-Nm
The name of the SOLA metadata template used by this web service.

Method-Nm
The name of the method that is associated with this web service.

Method-Ns
This is the method namespace, taken from the method tag on the SOAP request. It should be used when building a custom SOAP response.

Parent-Tag-Nm
For exits that may wish to append information to the outbound soap response, this represents the name of the parent tag to which the additional data should be appended. It is used as the parent tag when issuing the appendChild function of the SOLA Dom API.

User-Token
The user token, if any, that was included in the soap header of the inbound SOAP request (UserId for instance).

Dom-Ptr
This is the pointer/address that must be used when appending data to an existing SOLA soap response. It is used as the DOM handle when calling the SOLA DOM API. This variable is not available for use with SHI type exits.

Soap-Header-Ptr
For SHI type exits, this pointer is the address of the inbound soap header as it appears on the

soap request.

Soap-Header-Len

For SHI type exits, this is the length of the inbound soap header.

Soap-Req-Ptr

For SHI exits, this is a pointer to the inbound soap request.

Soap-Req-Len

The length of the inbound soap request.

Soap-Resp-Ptr

Not currently used.

Soap-Resp-Len

Not currently used.

Commarea-Ptr

For commarea programs, this will be the address of the legacy program's commarea (both on input and output).

Commarea-Len

Length of the legacy program's commarea.

Map-Ptr

Address of the current BMS map for a 3270 transaction.

Map-Len

Length of the BMS map.

Output Data: Passed back to SOLA from User Exits

Return-Code

Passed back to SOLA by the User Exit program. A value of zero indicates this SOLA should continue with normal processing.

A value of -1 indicates that SOLA should continue its normal fault processing. This is used by the Status (STA) user exit. The STA exit will be called near the end of normal SOLA processing or when SOLA has encountered and error and is about to throw a SOAP fault. If called when processing is ending, the 'Sola-Status' field will normally be set to 0 and the 'Sola-Status-Msg' field will consist of spaces. If SOLA has encountered a fault condition, the value of 'Sola-Status' will be -1 and 'Sola-Status-Msg' will contain the Soap Fault text that SOLA is about to throw. In the latter case, if the user exit program set the Return-Code to -1, then SOLA will continue to throw its normal Soap Fault as if the user exit was never called. Alternately, the user exit can set Return-Code to -2 in which case SOLA will instead throw a Soap Fault using the Fault Code, Fault String, and Fault Text provided by the user exit in the 'Error-Code', 'Fault-String', and 'External-Msg' fields respectively. If, during fault processing, the user exit sets the Return-Code to 0, SOLA will not throw a Soap Fault at all. Instead SOLA will send back a SOAP response containing whatever data it accumulated up to the time it began fault processing plus whatever has been appended to the SOAP response by the user exit.

A value of -3 indicates that the user exit program has composed its own SOAP response.
In this case the 'Custom-Resp-Ptr' field will be used by SOLA to send back the customized SOAP response.

Logging-Ind

This field is passed back to SOLA and indicates (in the case of fault processing) whether or not SOLA should log the error message in the SOLA error log.

Error-Code

The custom error code to use when formatting a custom fault message.

Fault-String

The custom fault string to use when formatting a custom fault message.

Internal-Msg

This message will be placed into the SOLA error log but will not be sent back to the client as part of fault processing.

External-Msg

This message will be sent back to the client as part of the soap fault.

Custom-Resp-Ptr

This is the Dom handle used by the user exit to create a document to be sent back to the client as a soap response. This soap response completely replaces SOLA's soap response/fault.

Custom-Resp-Len

Length of the custom soap response.

Legacy-Sysid

Instructs SOLA to DPL the legacy program request to the region indicated by this field.

Fault-Code

When instructing SOLA to send a custom SOAP fault, this field will indicate if the fault code should be Server or Client. Set the value of this field to 'S' for a server fault or 'C' for a client fault.

Using the API

Throwing a Custom SOAP Fault

To throw a custom SOAP fault from a user exit, follow these steps:

- Set the value of the 'Return-Code' variable to -2, which can be done by setting the 'Throw-Custom-Fault' 88 level to TRUE.
- Set the desired value for the 'Error-Code' field.
- Move the text that is to appear in the fault string into the 'Fault-String' field.
- Move the text that is to appear in the fault message into the 'External-Msg' field.
- If desired, move a diagnostic message into the 'Internal-Msg' field. This will appear in the SOLA error log providing you set the value of the 'Logging-Ind' to 'Y'.
- Pass control back to SOLA.

Sending a Custom Soap Response

To instruct SOLA to stop processing and send a soap response of your choosing, follow these steps:

- Set the value of the WS-DOM-HANDLE to null (low-values). This is necessary because you will be creating a completely new document.
- Build a new document using the SOLA DOM API. Start by executing the 'createDocument' function followed by 'appendChild', 'setAttribute', etc. as desired to create your custom soap document.
- Execute the SOLA DOM API 'finalize' function.
- Set the value of 'Return-Code' to -3, which can be done by setting the 'Custom-Response' 88 level to TRUE.
- Pass control back to SOLA

Appending Data to SOLA's SOAP Response

To append data to a response that has been built by SOLA you will need to use the same DOM API handle used by SOLA to build the SOAP response.

- First check the value of 'Sola-Status'. This will inform you whether SOLA has encountered a condition under which it would normally send a soap fault. If the value is zero then SOLA is about to send back a normal soap response to the requestor and you will have the opportunity to append data to that response or instruct SOLA to send a custom soap fault as described above. If the value is -1 then SOLA is about to throw a soap fault. In this case you can instruct SOLA to continue with its own fault processing by setting 'Return-Code' to -1, instruct SOLA to send a custom SOAP fault as described above, or append data to the soap response and deliver the altered response instead of throwing a soap fault.
- SOLA will provide the DOM pointer/handle that should be used when executing SOLA DOM API functions. This will be contained in the 'Dom-Ptr' field. The WS-DOM-HANDLE field (see DOM API documentation) should be set to this value. By using the same pointer/handle you will be appending your data to the proper soap response.
- SOLA will provide the user exit with a parent tag name appropriate for appending additional data. This parent tag name will be specified in field 'Parent-Tag-Nm'.
- Perform various DOM API functions ('appendChild' etc.) to add the desired XML constructs to the soap response.
- Make sure to set the value of 'Return-Code' to zero and return control to SOLA.

Setting SysId and TranId Without a User Exit

You can specify the SysId and TranId without using a user exit by passing them in the SOAP header as follows (case sensitive):

```
<soap:Header>
   <SysId>yourSysId</SysId>
   <TranId>yourTranId</TranId>
   …
```

If the SysId and TranId values are also passed by a user exit, the user exit values will override the value passed in the header.

Sample Program

This sample program is of type Soap Header In. In this example, the program will extract from the soap request the User Id and an account number. From this it will look up the account balance and apply business rules that determine if this particular user is allowed to perform the request activity ‘accountUpdate’ on accounts with a balance greater than $10,000,000.

This sample program can be found in your SAMPLIB under the name EXITPROG.

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. EXAMPLE.
000300 ENVIRONMENT DIVISION.
000400 DATA DIVISION.
000500*----------------------------------------------------------------*
000501* Note that the business rules, DB2 tables, etc. contained in
000502* this example are all fictitious
000510*----------------------------------------------------------------*
000600 WORKING-STORAGE SECTION.
00060475
000700*----------------------------------------------------------------*
000800
002900 01  WS-MISC-DISPLAY-DATA.
002901     05  WS-ERROR-MSG               PIC  X(100) VALUE SPACES.
002910
002920 01  WS-SWITCHES.
003000     05  WS-TAG-FOUND-SW            PIC S9(04) VALUE +0 BINARY.
003100         88  TAG-NOT-FOUND          VALUE +0.
003200         88  TAG-FOUND              VALUE +1.
003300         88  Dom-Error              VALUE -1.
003500
003510*    The following copy book contain all working storage
003520*    variables needed by the SOLA DOM API program.
003530
003600     COPY XMLDOMWS.
003700
003701     EXEC SQL
003702        INCLUDE SQLCA
003703     END-EXEC.
003704
003705     EXEC SQL
003706        INCLUDE TBACTUSR
003707     END-EXEC.
003708
003709     EXEC SQL
003711        INCLUDE TBACTBAL
003712     END-EXEC.
003713
003800*----------------------------------------------------------------*
003900 LINKAGE SECTION.
00932014
004000*----------------------------------------------------------------*
```

```
004100 01  DFHCOMMAREA.
004200     COPY SOLAEXIT REPLACING ==:WS-:== BY ==LK-==.
004300
004400 01  LK-SOAP-REQUEST              PIC  X(01).
004500 01  LK-SOAP-HEADER               PIC  X(1000).
004600
004700     COPY XMLCNTLB.
004800
004900*----------------------------------------------------------------*
005000 PROCEDURE DIVISION.
005100*----------------------------------------------------------------*
005200     CONTINUE.
005300*----------------------------------------------------------------*
005400 0000-MAINLINE.
005500*----------------------------------------------------------------*
005600
005700     PERFORM 0010-INITIALIZE
005800        THRU 0010-EXIT
005900
006300     PERFORM 1000-AUTHORIZE-REQUEST
006400        THRU 1000-EXIT
006500
007600     GO TO 9999-RETURN
007700
007800     CONTINUE.
007900
008000 0000-EXIT.
008100     EXIT.
008200
008300*----------------------------------------------------------------*
008400 0010-INITIALIZE.
008500*----------------------------------------------------------------*
008600
008601     SET LK-Normal-Continue TO TRUE
008602
008610     IF LK-Soap-Header-In AND
008700        LK-Method-Nm = 'accountUpdate'
008800        CONTINUE
008900     ELSE
009000        GO TO 9999-Return
009100     END-IF
009200
009210     IF LK-Soap-Header-Len > +0
009220        CONTINUE
009230     ELSE
009231        MOVE 'SOAP Header not found' TO WS-ERROR-Msg
009240        PERFORM 8000-FORMAT-BAD-REQ-FAULT
009241           THRU 8000-EXIT
009242        GO TO 9999-Return
009250     END-IF
009260
009300     SET WS-DOM-HANDLE TO LK-Dom-Ptr
009500
009600     CONTINUE.
009700
```

```
009800 0010-EXIT.
009900     EXIT.
010000
010100*---------------------------------------------------------------*
010110 1000-AUTHORIZE-REQUEST.
010300*---------------------------------------------------------------*
010400
012100     PERFORM 7000-PARSE-SOAP-REQUEST
012200        THRU 7000-EXIT
012300
012310     IF DOM-ERROR
012311        MOVE 'Error parsing soap request' TO WS-ERROR-MSG
012312        PERFORM 8100-FORMAT-DOM-ERROR
012313           THRU 8100-EXIT
012314        GO TO 1000-EXIT
012315     END-IF
012320
012400     SET  WS-DOM-GET-ELEMENT TO TRUE
012500     MOVE +1                 TO WS-DOM-SEQ
012700     MOVE +0                 TO WS-Dom-Rc
012800     MOVE 'UserId'           TO WS-DOM-TAG-NAME
012900     MOVE SPACES             TO WS-DOM-Value
013000     MOVE ZERO               TO WS-DOM-VALUE-LENGTH
013100
013200     PERFORM 7100-CALL-DOM-API
013300        THRU 7100-EXIT
013400
013410     EVALUATE TRUE
013500        WHEN TAG-FOUND
014200           CONTINUE
014510        WHEN TAG-NOT-FOUND
014511           MOVE 'User Id not found' TO WS-ERROR-MSG
014513           PERFORM 8000-FORMAT-BAD-REQ-FAULT
014514              THRU 8000-EXIT
014515           GO TO 1000-EXIT
014520        WHEN DOM-ERROR
014521           MOVE 'Error parsing UserId' TO WS-ERROR-MSG
014522           PERFORM 8100-FORMAT-DOM-ERROR
014523              THRU 8100-EXIT
014524           GO TO 1000-EXIT
015400     END-EVALUATE
015500
015600     MOVE +0              TO WS-Dom-Rc
015700     MOVE 'AccountNumber' TO WS-DOM-TAG-NAME
015800     MOVE SPACES          TO WS-DOM-Value
015900     MOVE ZERO            TO WS-DOM-VALUE-LENGTH
016000
016100     PERFORM 7100-CALL-DOM-API
016200        THRU 7100-EXIT
016300
016400     EVALUATE TRUE
016500        WHEN TAG-FOUND
016600           CONTINUE
016700        WHEN TAG-NOT-FOUND
016800           MOVE 'Account number not found' TO WS-ERROR-MSG
```

```
016810           PERFORM 8000-FORMAT-BAD-REQ-FAULT
016820              THRU 8000-EXIT
016821           GO TO 1000-EXIT
016900        WHEN DOM-ERROR
016910           MOVE 'Error parsing AccountNumber' TO WS-ERROR-MSG
016920           PERFORM 8100-FORMAT-DOM-ERROR
016930              THRU 8100-EXIT
016940           GO TO 1000-EXIT
017100     END-EVALUATE
017200
017300     EXEC SQL
017400        SELECT USER_AUTH_LVL
017500          INTO :UAT-USER-AUTH-LVL
017600          FROM TBACTUSR
017700         WHERE USER_ID = :UAT-USER-ID
017800     END-EXEC
017900
017910     EVALUATE SQLCODE
017920        WHEN +0
017921           CONTINUE
017930        WHEN +100
017931           MOVE 'User not on file' TO WS-ERROR-MSG
017932           PERFORM 8300-FORMAT-APPLICATION-ERROR
017933              THRU 8300-EXIT
017934           GO TO 1000-EXIT
017940        WHEN OTHER
017941           MOVE 'Error selecting user auth level' TO WS-ERROR-MSG
017942           PERFORM 8200-FORMAT-DB2-ERROR
017943              THRU 8200-EXIT
017944           GO TO 1000-EXIT
017950     END-EVALUATE
017960
018000     EXEC SQL
018100        SELECT ACCOUNT_BALANCE
018200          INTO :ACT-ACCOUNT-BALANCE
018210          FROM TBACTBAL
018220         WHERE ACCOUNT_NUMBER = :ACT-ACCOUNT-NUMBER
018230     END-EXEC
018240
018241     EVALUATE SQLCODE
018242        WHEN +0
018243           CONTINUE
018244        WHEN +100
018245           MOVE 'Accout number not on file' TO WS-ERROR-MSG
018246           PERFORM 8300-FORMAT-APPLICATION-ERROR
018247              THRU 8300-EXIT
018248           GO TO 1000-EXIT
018249        WHEN OTHER
018250           MOVE 'Error selecting account info' TO WS-ERROR-MSG
018251           PERFORM 8200-FORMAT-DB2-ERROR
018252              THRU 8200-EXIT
018253           GO TO 1000-EXIT
018254     END-EVALUATE
018255
018256*    In this example assume that there are three levels of
```

```
018257*    user authorization. 'L' - Low, 'M' - Medium, and 'H'- High.
018258*    High is required by the business to update an account whose
018259*    balance is greater than $10,000,000.  Thus...
018260
018261     IF ACT-ACCOUNT-BALANCE > 10000000.00 AND
018262        UAT-USER-AUTH-LVL NOT = 'H'
018263
018264        MOVE 'Insufficient auth level' TO WS-ERROR-MSG
018265
018270        PERFORM 8300-FORMAT-APPLICATION-ERROR
018280           THRU 8300-EXIT
018281
018290     END-IF
018300
018400     CONTINUE.
018500
018600 1000-EXIT.
018700     EXIT.
018800
018900*----------------------------------------------------------------*
019000 7000-PARSE-SOAP-REQUEST.
019100*----------------------------------------------------------------*
019200
019300     SET  WS-DOM-PARSE               TO TRUE
019400     MOVE LK-SOAP-REQ-LEN            TO WS-Dom-VALUE-LenGTH
019500     SET  ADDRESS OF LK-SOAP-REQUEST TO LK-SOAP-REQ-PTR
019600
019700     CALL WS-Dom-Api USING WS-Dom-Rc
019800                           WS-Dom-Msg
019900                           WS-DOM-HANDLE
020000                           WS-Dom-Function
020100                           WS-Dom-Parent
020200                           WS-DOM-TAG-NAME
020300                           LK-SOAP-REQUEST
020400                           WS-Dom-VALUE-LenGTH
020500
020600     IF WS-DOM-RC NOT = +0
020610        SET DOM-ERROR TO TRUE
021400     END-IF
021500
021600     CONTINUE.
021700
021800 7000-EXIT.
021900     EXIT.
022000
045200*----------------------------------------------------------------*
045300 7100-CALL-DOM-API.
045400*----------------------------------------------------------------*
045500
045600     CALL WS-Dom-Api USING WS-Dom-Rc
045700                           WS-Dom-Msg
045800                           WS-DOM-HANDLE
045900                           WS-Dom-Function
046000                           WS-Dom-Parent
046100                           WS-DOM-TAG-NAME
```

```
046200                           WS-Dom-Value
046300                           WS-Dom-VALUE-LenGTH
046400
046500     EVALUATE WS-DOM-RC
046600        WHEN +0
046700           SET TAG-FOUND     TO TRUE
046800
046900        WHEN +4
047000           SET TAG-NOT-FOUND TO TRUE
047100
047200        WHEN OTHER
047210           SET DOM-ERROR TO TRUE
048000
048100     END-EVALUATE
048200
048300     CONTINUE.
048400
048500 7100-EXIT.
048600     EXIT.
048700
048701*----------------------------------------------------------------*
048702 8000-FORMAT-BAD-REQ-FAULT.
048703*----------------------------------------------------------------*
048704
048705     SET  LK-Throw-Custom-Fault  TO TRUE
048706     SET  LK-Log-Internal-Msg    TO TRUE
048707     MOVE -1001                  TO LK-Error-Code
048709     MOVE 'Application Error'    TO LK-Fault-String
048710     MOVE 'Invalid SOAP Request' TO LK-External-Msg
048711     MOVE WS-ERROR-MSG           TO LK-Internal-Msg
048714
048715     CONTINUE.
048716
048717 8000-EXIT.
048718     EXIT.
048719
048720*----------------------------------------------------------------*
048721 8100-FORMAT-DOM-ERROR.
048722*----------------------------------------------------------------*
048730
048731     SET  LK-Throw-Custom-Fault TO TRUE
048732     SET  LK-Log-Internal-Msg   TO TRUE
048733     MOVE WS-Dom-Rc             TO LK-Error-Code
048734     MOVE 'Dom API Error'       TO LK-Fault-String
048735     MOVE WS-ERROR-MSG          TO LK-External-Msg
048736     MOVE WS-Dom-Msg            TO LK-Internal-Msg
048740
048750     CONTINUE.
048760
048770 8100-EXIT.
048780     EXIT.
048790
048791*----------------------------------------------------------------*
048792 8200-FORMAT-DB2-ERROR.
048793*----------------------------------------------------------------*
```

```
048794
048795     SET  LK-Throw-Custom-Fault TO TRUE
048796     SET  LK-Log-Internal-Msg   TO TRUE
048797     MOVE SQLCODE               TO LK-Error-Code
048798     MOVE 'DB2 Error'           TO LK-Fault-String
048799     MOVE WS-ERROR-MSG          TO LK-External-Msg
048800     MOVE WS-Dom-Msg            TO LK-Internal-Msg
048801
048802     CONTINUE.
048803
048804 8200-EXIT.
048805     EXIT.
048806
048807*---------------------------------------------------------------*
048808 8300-FORMAT-APPLICATION-ERROR.
048809*---------------------------------------------------------------*
048810
048811     SET  LK-Throw-Custom-Fault TO TRUE
048812     SET  LK-Log-Internal-Msg   TO TRUE
048813     MOVE -1002                 TO LK-Error-Code
048814     MOVE 'Application Error'   TO LK-Fault-String
048815     MOVE WS-ERROR-MSG          TO LK-External-Msg
048817
048818     CONTINUE.
048819
048820 8300-EXIT.
048821     EXIT.
048822
048830*---------------------------------------------------------------*
048900 9999-RETURN.
049000*---------------------------------------------------------------*
049100
049200     GOBACK
049300
049400     CONTINUE.
049500
049600 9999-EXIT.
049700     EXIT.
```

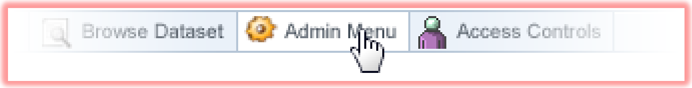
Running User Exits in the SOLA IMS Container (SOLA STC)

We recommend that you APF authorize the SOLA load library when you're running the SOLA STC. This will allow you to validate Security credentials passed through a SOAP request against the SAF interface.

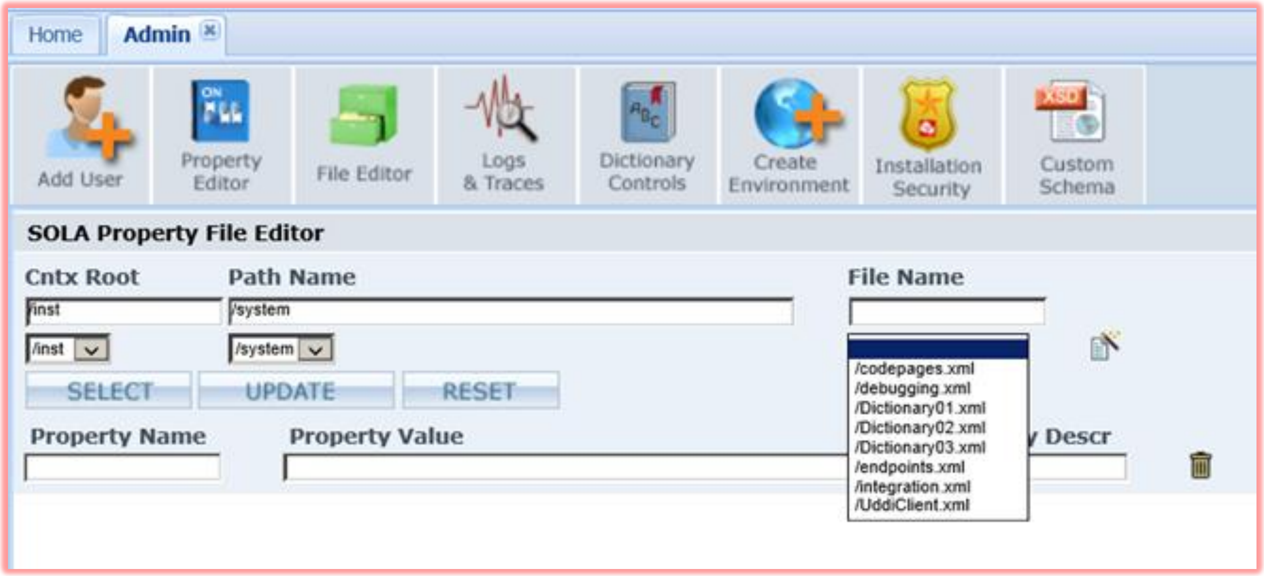
User exits are loaded from the STEPLIB DD. If you've configured SOLA Exits then the SOLA Exit load library must be added to the //STEPLIB DD of the SOLA STC Job.

- If you're running the SOLA STC in Authorized mode (the SOLA loadlib is APF authorized) then the SOLA Exit library must also be authorized.

Property Changes



To make changes to SOLA system properties, access the SOLA Admin Menu screen by selecting **SOLA Administration Admin Menu** from the SOLA Button Bar.



SOLA Property File Editor

Cntx Root: /inst (menu: /inst)

Path Name: /system (menu: /system)

File Name:

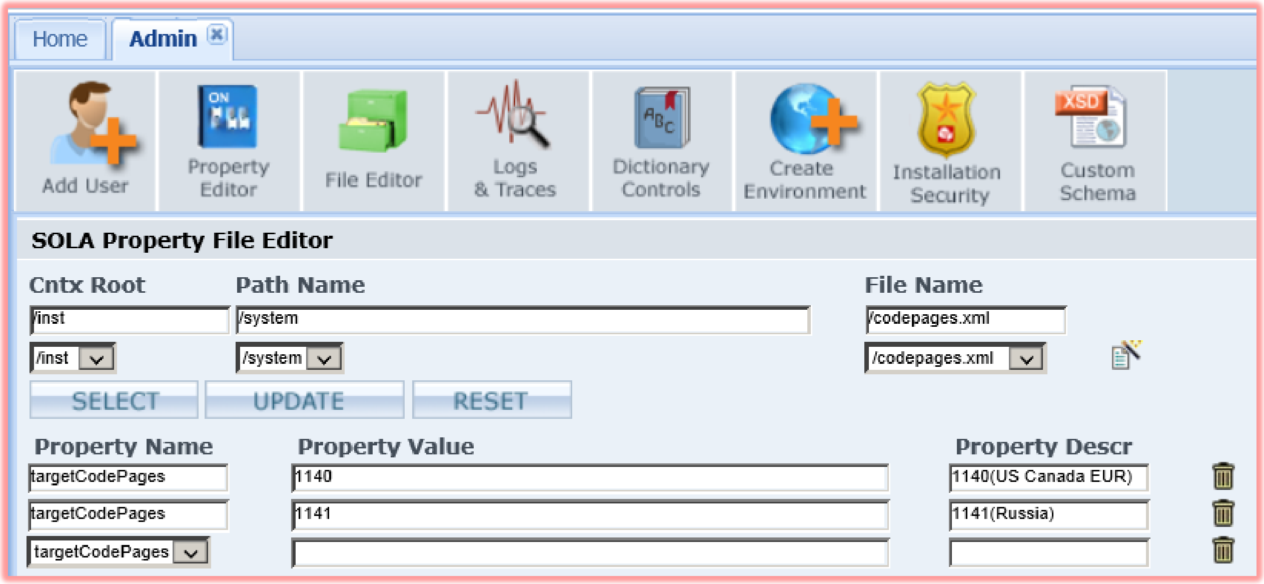
/codepages.xml
/debugging.xml
/Dictionary01.xml
/Dictionary02.xml
/Dictionary03.xml
/endpoints.xml
/integration.xml
/UddiClient.xml

Property Name | Property Value | y Descr

Click the **Property Editor** icon to display the Property File Editor screen set to Property Editor Mode (this is the default selection).

To edit system properties, you must first select a file that contains the properties you wish to work with. Enter or browse for the file's path using the **Cntx Root**, **PathName** and **FileName** fields or menus (fields to enter, menus to browse).

When you have selected a file, the properties in that file will be displayed under the **Property Name**, **Property Value** and **Property Descr** fields.



SOLA Property File Editor

Cntx Root	Path Name	File Name
/inst	/system	/codepages.xml
/inst	/system	/codepages.xml

Property Name	Property Value	Property Descr
targetCodePages	1140	1140(US Canada EUR)
targetCodePages	1141	1141(Russia)
targetCodePages		

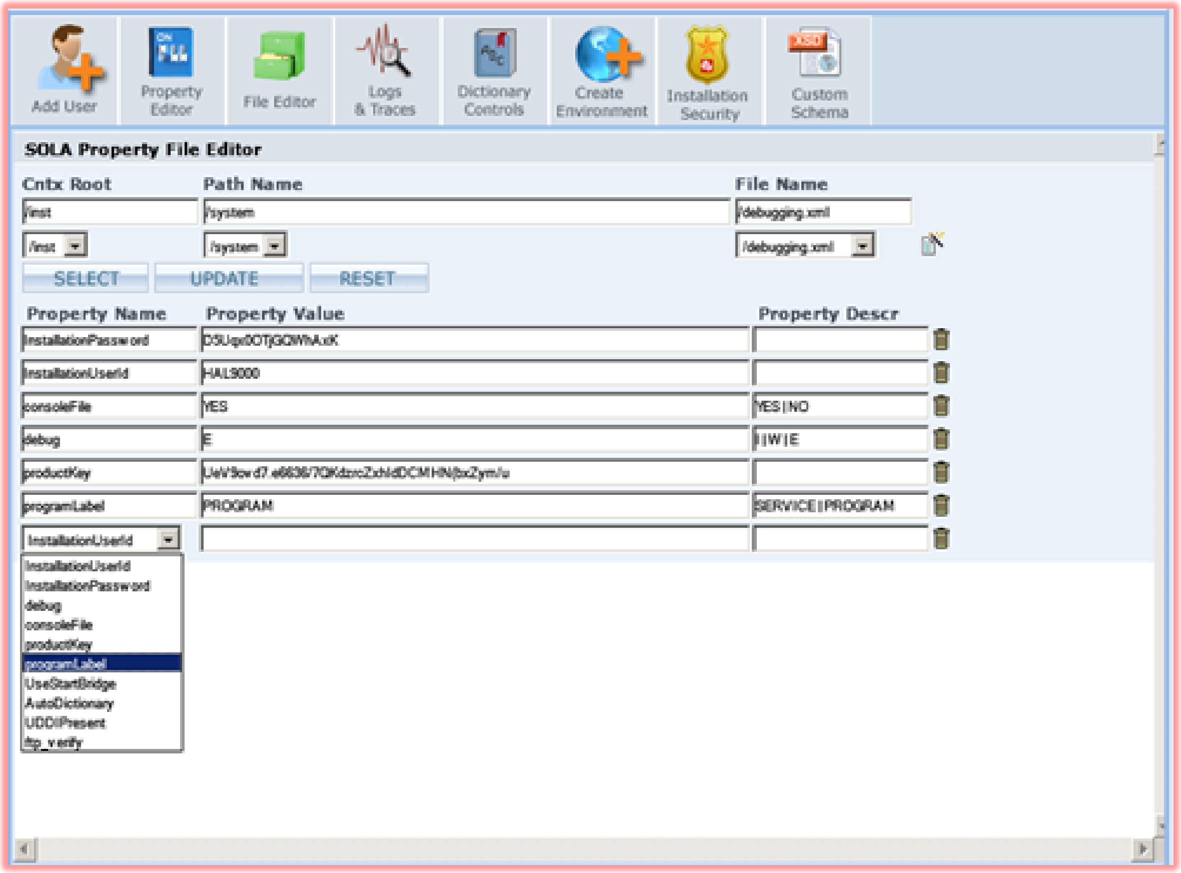
You can make changes to existing properties and their values, or you can enter a new property in a blank field.

To add additional fields, click the 📝 button. To remove an unwanted blank field or to delete a property and its value, click the 🗑 button.

The allowable Property name and value for the file **codepages.xml** is:

targetCodePages	Leave blank to use EBCDIC default code page 37 **OR** enter any valid 4 digit code page. Ex: 1140 to use the Euro currency update of code page/CCSID 37. The Property Value AND Property Description must be entered.

SOLA Property File Editor – **debugging.xml** illustration:



SOLA Property File Editor

Cntx Root	Path Name	File Name
/inst	/system	/debugging.xml
/inst	/system	/debugging.xml

Property Name	Property Value	Property Descr
InstallationPassword	D5Uqx0OTjGQWhAxK	
InstallationUserId	HAL9000	
consoleFile	YES	YES|NO
debug	E	I|W|E
productKey	UeV9owd7.e6636/7QKdzrcZxhldDCMHN/bxZym/u	
programLabel	PROGRAM	SERVICE|PROGRAM
InstallationUserId		

InstallationUserId
InstallationPassword
debug
consoleFile
productKey
programLabel
UseStartBridge
AutoDictionary
UDDIPresent
ftp_verify

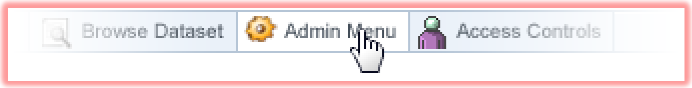
The allowable Property names and values for the file **debugging.xml** are listed below:

Property	Allowable Values
InstallationUserid	Optional. Enter a valid SAF ID. SOLA will use this ID to run all back end transactions that support the IDE.
InstallationPassword	An encrypted value is pre-filled here. Do not modify the installation password on this page. Changing the installation password is done on installation default setup page14.
debug	I - Log all informative, warnings & Errors W - Log all warnings and errors messages E - Log only errors Messages

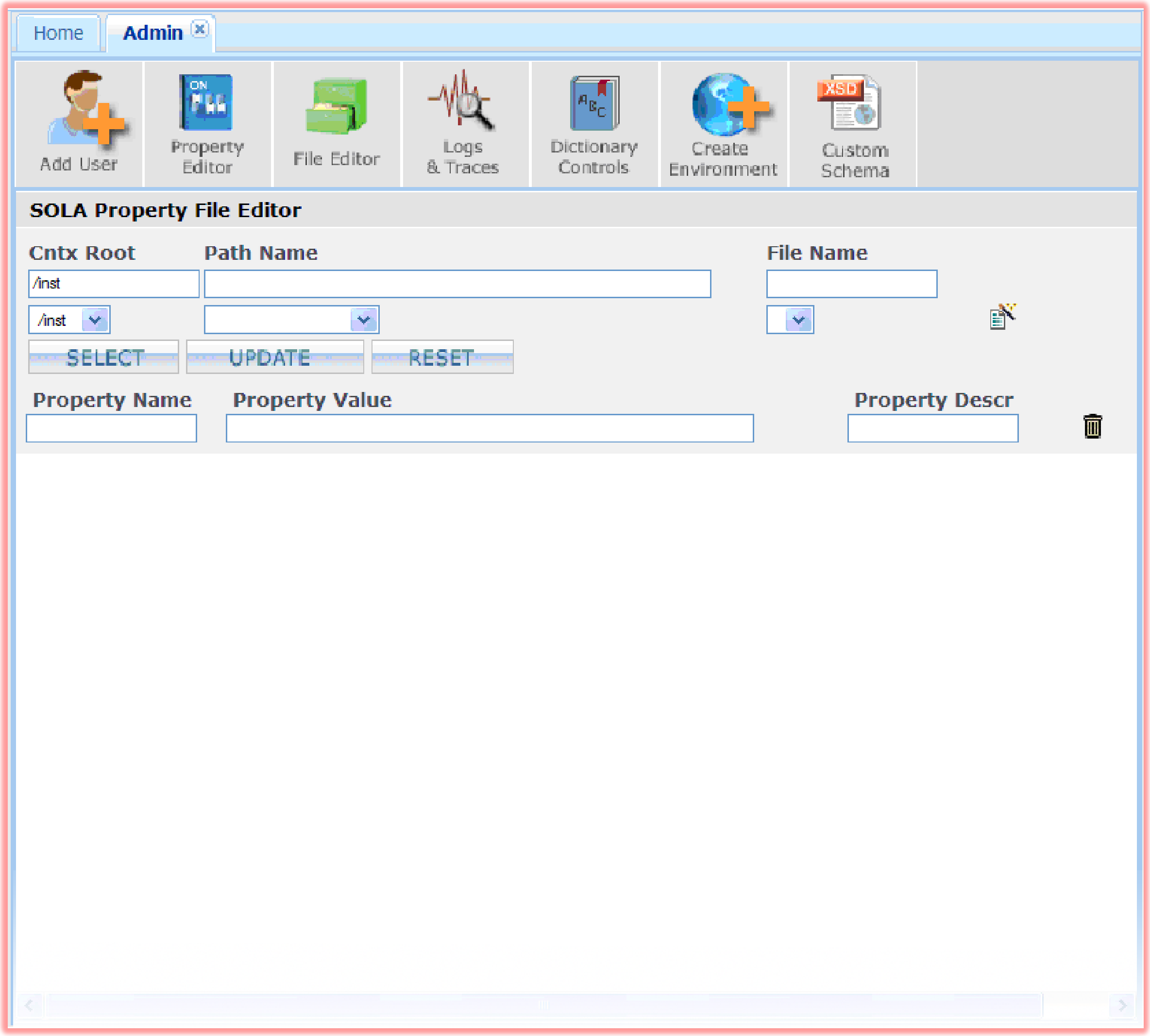
	Value "E" is default and recommended.
consoleFile	YES – creates and appends file in logs folder with error or warning messages. NO – sends messages to the server console.
productKey	SOLA's valid product key. Use the product key that was provided to you separately.
LMIntegratedMode	Enables SOLA-Lifecycle Manager integration by entering 'Y'.
exclude_level88	When set to 'Y' all 88 level items will be excluded from processing during IMPORT.
logRetentionDays	The number of days the Logs/Traces are to be retained; the default is 999.
UseStartBridge	SOLA CICS Container only. If you have PTF SOFX571 applied for module XMLPC400, you can disable LinkBridgeSupport during analysis. Using LinkBridgeSupport during analysis can cause problems in some installations. Y - Disable LinkBridgeSupport. N - Do not disable LinkBridgeSupport.
AutoDictionary	Y – Whenever a user changes the name of a variable during analysis, SOLA will capture both the original name and what it was changed to and store them in the SOLA Dictionary. N – SOLA will not capture user changes (of variable names), and all entries in the SOLA dictionary must be added manually.
UDDIPresent	If the UDDIPresent property in debugging.xml is set to 'Y' the UDDIClient.xml file will be read. This file contains all of the properties that the client SDK expects to be present in a uddiclient.properties file a well as other parameters such as UDDIServerAddress, User and Password. If the UDDIPresent property is set to 'N'

ftp_verify	The default value of this property is 'N'. This property allows customers to control how ‘user logons’ to the IDE are verified against RACF or equivalent. By default the SOLA mainframe runtime processes the logon request. For SOLA CICS Container it is recommended you leave the default as 'N'. For SOLA IMS Container (Started Task) adjust the value as follows: ○ If a started task is configured to run in Authorized state [Refer to "Chapter 3: Customizing SOLA IMS Container on a z/Series mainframe", Section: APF AUTHORIZE THE SOLA LOAD LIBRARY (Recommended)] then leave the property to default 'N'. ○ If a started task is configured to run in Unauthorized state then set the value to 'Y' so the FTP mechanism will be used to verify user credentials at logon.
IDEMainframeProxy	
IDEAuthPerRequest	
IDEServer	

System File Changes



To make changes to SOLA system files, access the SOLA Admin Menu screen by selecting **SOLA Administration** from the SOLA Home page.

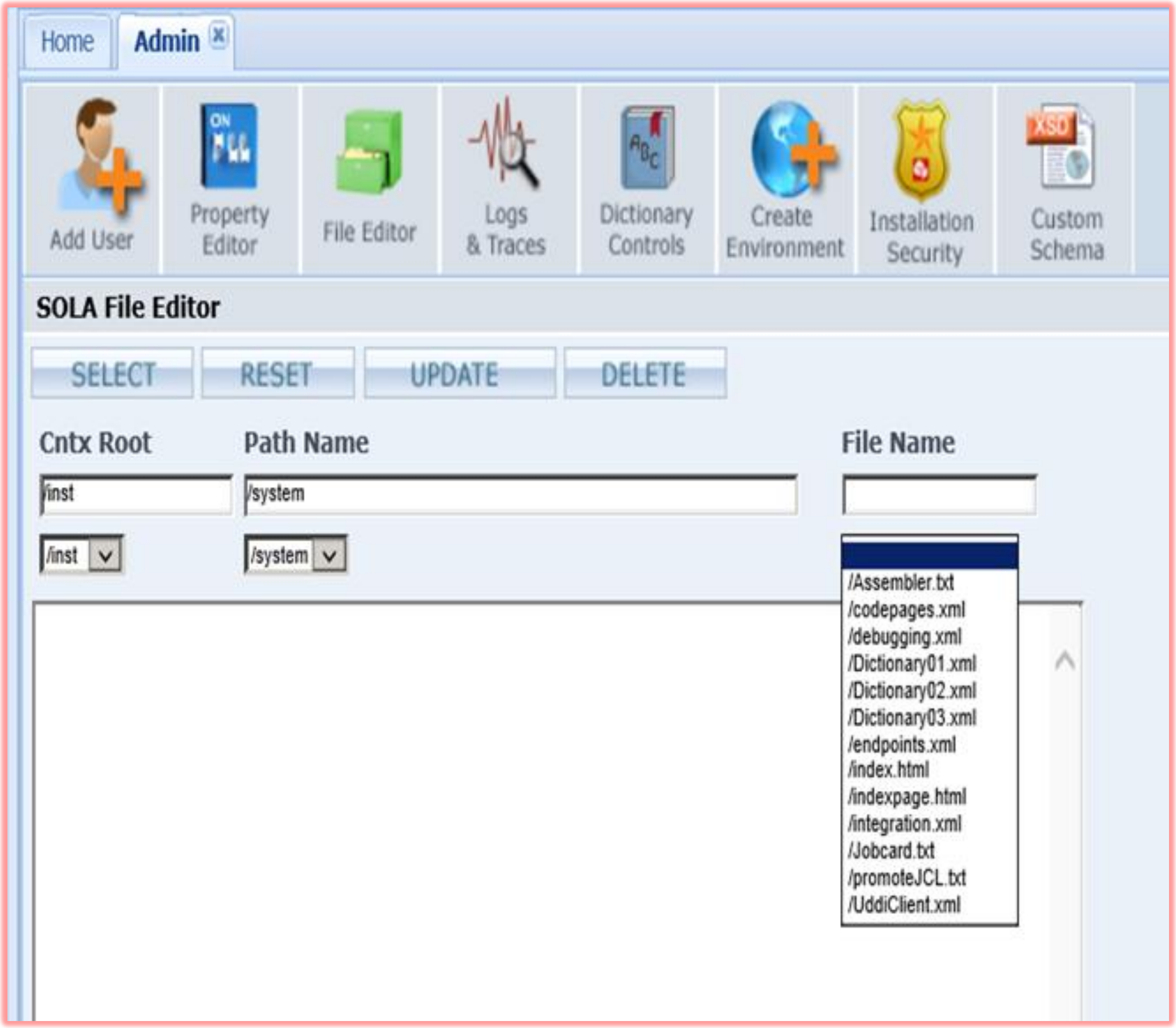


SOLA Property File Editor

Cntx Root	Path Name	File Name
/inst		
/inst		

Property Name	Property Value	Property Descr

Click on the **File Editor** to display the Property File Editor screen set to File Editor mode.



/inst

/system

/Assembler.txt
/codepages.xml
/debugging.xml
/Dictionary01.xml
/Dictionary02.xml
/Dictionary03.xml
/endpoints.xml
/index.html
/indexpage.html
/integration.xml
/Jobcard.txt
/promoteJCL.txt
/UddiClient.xml

To edit a system file, enter or browse for its path using the **Cntx Root**, **PathName** and **FileName** fields or menus (fields to enter, menus to browse).

Within the **Cntx Root** of /Inst and **PathName** of /system there are several files:

- **index.html:** the html for the SOLA home page. Customizing the SOLA home page is discussed in section Customization of the SOLA Home Page on page 9.
- **Codepages.xml:** this file can be updated to add EBCDIC codepages that are relevant to the mainframe applications at your site. The default is IBM US EBCDIC CCSID 37. If applications need to process "€" symbol then setup CCSID 1140 in this file. CCSID 1140 is the Euro currency update of code page/CCSID 37. In that code page, the "¤" (currency sign) character at code point x' 9F' is replaced with the "€" (Euro sign)

character. While service enabling mainframe programs, the captured codepages in this property file will be available to SOLA users for selection.

- **Debugging.xml:** updating debugging.xml is discussed in the Property Changes section on page 41.
- **Endpoints.xml:** updating the endpoints.xml file is discussed below.
- **UddiClient.xml:** this file will be read whenever the UDDIPresent in debugging.xml is set to "Y" (see above). It contains all of the properties that the client SDK normally expects to be present in a uddiclient.properties file. It also contains connection parameters such as address, user id and password.

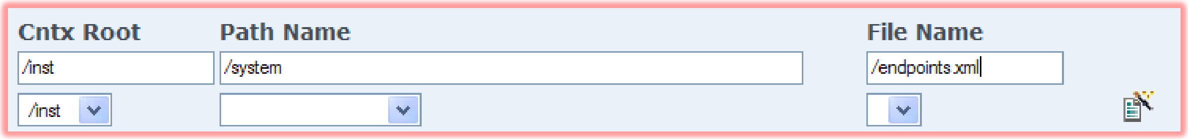
When you have selected a file, its contents will appear in the large field below the path menus (you can scroll if necessary). Make whatever changes you want, then click **UPDATE**.

Updating the endpoints.xml File

In order to administer the endpoints that SOLA will manage, you will need to enter those endpoints into the endpoints.xml file. You can update the file using either the Property Editor or the File Editor. The File Editor will display the entire contents of the file, whereas the property editor will isolate the properties, their values and their descriptions. It is recommended that you use the Property Editor to enter endpoints into the endpoints.xml file.

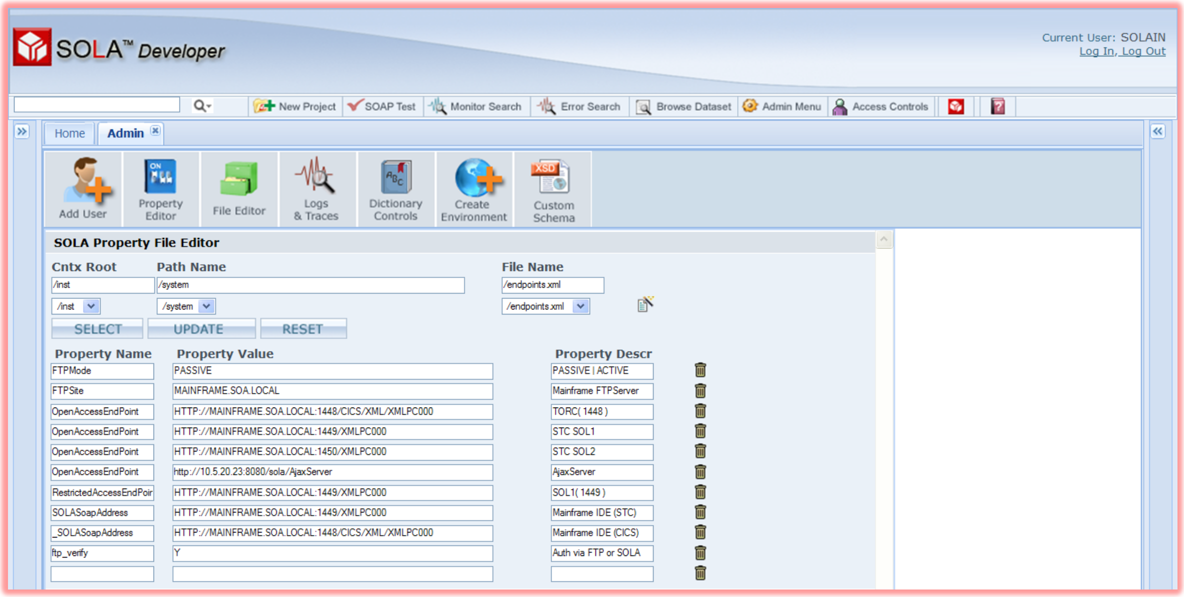
To access the Property Editor, follow the instructions in the
Property Changes section on page 40.

Fill in the **CntxRoot**, **PathName** and **FileName** fields with the following:



Cntx Root	Path Name	File Name
/inst	/system	/endpoints.xml
/inst		

As shown in the screen shot below, you can use the drop down menus to select the path to the file.



SOLA Property File Editor

Cntx Root	Path Name	File Name
/inst	/system	/endpoints.xml
/inst	/system	/endpoints.xml

Property Name	Property Value	Property Descr
FTPMode	PASSIVE	PASSIVE | ACTIVE
FTPSite	MAINFRAME.SOA.LOCAL	Mainframe FTPServer
OpenAccessEndPoint	HTTP://MAINFRAME.SOA.LOCAL:1448/CICS/XML/XMLPC000	TORC(1448)
OpenAccessEndPoint	HTTP://MAINFRAME.SOA.LOCAL:1449/XMLPC000	STC SOL1
OpenAccessEndPoint	HTTP://MAINFRAME.SOA.LOCAL:1450/XMLPC000	STC SOL2
OpenAccessEndPoint	http://10.5.20.23:8080/sola/AjaxServer	AjaxServer
RestrictedAccessEndPoir	HTTP://MAINFRAME.SOA.LOCAL:1449/XMLPC000	SOL1(1449)
SOLASoapAddress	HTTP://MAINFRAME.SOA.LOCAL:1449/XMLPC000	Mainframe IDE (STC)
_SOLASoapAddress	HTTP://MAINFRAME.SOA.LOCAL:1448/CICS/XML/XMLPC000	Mainframe IDE (CICS)
ftp_verify	Y	Auth via FTP or SOLA

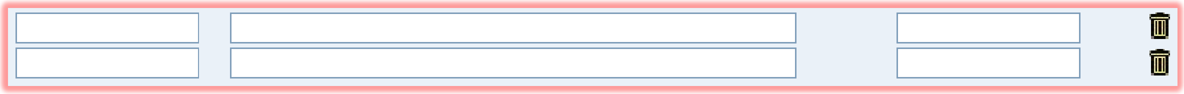
The key-value pairs will change to reflect the values in the endpoints.xml file. The following is an explanation of the properties in the file along with allowable values.

NOTE: SOLA FTPServer now supports Explicit-SSL to transfer artifacts to the mainframe. To setup FTPServer as SSL, set attribute **FTPSite**. "**FTPSite**" property can be set as "**ftpes**:<MainframeHostName>" or "**ftpes**:<MainframeHostName>:<ftpPort>"

- **FTPMode:** ACTIVE or PASSIVE. Determines which FTP mode the SOLA IDE will use for communication with the mainframe FTP server for retrieving datasets. ACTIVE is recommended. Set "FTPMode" as "PASSIVE" if you want to use FTP over SSL.
- **FTPSite:** The SOLA IDE communicates with z/OS in two ways – via SOLA web services for all directory access and via FTP for access to z/OS datasets, JES output and the JES scheduler. In this value you need to specify the address of the z/OS FTP server.
- **OpenAccessEndPoint:** The FQDN of a SOLA Container. This container is where developers test their services. This container may or may not be the same as the SOLASoapAddress FQDN.
- **RestrictedAccessEndPoint:** The FQDN of a secured SOLA Container. This container is where services are executed. Only administrators will be able to see this value in an endpoint drop-down. This value is optional.
- **SOLASoapAddress:** The FQDN of a SOLA Container. This container provides the mainframe backend for the SOLA IDE. This container may or may not be the same as the OpenAccessEndPoint FQDN.

The key-value pair contents of the endpoints.xml file are pre-filled. All unknown values are not filled or filled with value "InActive".

If you do not intend to use some values, don't leave them blank. They must be removed. To remove blank or unused rows, click on the 🗑 button associated with the row(s) you want to remove.



Click the **Update** button then **OK** to confirm in order to update the file and propagate the changes.

Updating the UddiClient.xml File

The UddiClient.xml file will be read whenever the UDDIPresent property in debugging.xml is set to “Y”. This file contains all of the properties that the client SDK normally expects to be present in a uddiclient.properties file. The following is a list of properties present in the file:

- **UDDIServerAddress:** The FQDN of a central UDDI V3 registry for publishing SOLA Services. Release 6.0 of SOLA supports Policy Manager from SOA Software.
- **UDDIUser:** The UserId to use to connect to the UDDIServer.
- **UDDIPassword:** The password associated with the UDDI UserId.
- **DEBUG.LEVEL:** the desired debug level, from 0 (no logging) to 400 (full logging), in increments of 100.
- **LOG.INFO:** this setting determines whether informational messages will be logged. Set to true or false.
- **LOG.TRACE:** this setting determines whether trace messages will be logged. Set to true or false.
- **LOG.WARNING:** this setting determines whether warning messages will be logged. Set to true or false.
- **LOG.ERROR:** this setting determines whether error messages will be logged. Set to true or false.
- **LOG.HANDLERS.File.Filename:** the path to the log file.
- **LOG.HANDLERS.File:** the desired log handler for logging to a file system.
- **LOG.Immediate:** a Y or N value that determines whether the logs should be buffered or sent to the log handlers immediately.
- **REFRESH.com.digev.fw.config. file.FileConfig:** this setting determines the configuration file refresh interval.
- **jmx.domain.name:** domain name for JMX connectivity.

Security

Overview

Build-Time Security

Build-Time security refers to all SOLA features accessible through the SOLA IDE.

SOLA recognizes three levels of user access; administrator, programmer and public access. Furthermore, there are two categories of administrators with escalating privileges.

The SOLA IDE contains a 'login' feature, though it is not necessary to login in order to use some of the features of the IDE. Using SOLA without logging in will only provide access to public access features. Whenever a restricted feature is accessed by a user who is not logged in, SOLA will prompt the user for a username and password.

The three levels of user access are as follows:

- **Administrator:** consists of two categories, whose escalating privileges are explained in detail in Chapter 1: Authorization Overview. These categories are:
  - **SOLA Administrator:** has access to all administration features.
  - **Project Administrator:** has access to project and method-level administration features.

- **Programmer:** this type of user can access all of SOLA's functional features, but lacks access to security and administration options. This user can import and analyze programs and resources in projects in which they are authorized to work.

- **Public Access:** this is the lowest access level and can do the following:
  - Directory search
  - View WSDL
  - Quick Test function pointing to test containers

Runtime Security

SOLA provides the following three options for runtime security:

- WS Security
    - UsernameToken
    - X509 Token
    - XML Encryption
    - XML Signature
- Channel and Resource Lockdown
- Customized security using security exits (programs).

Security features are configurable via a defined security policy for an operation or the entire installation.

SOLA Analyzers

Note: This feature is not available in a SOLA IMS Container.

By default, the SOLA CICS Container runs with the default CICS Analyzer DFHWBADX. With this analyzer, which runs under transaction CWXN, it is not possible to run your user transactions under a SAF ID. If, for accounting or authorization reasons, you need to run your user transactions under a SAF ID that's passed up in the SOAP message, you will need to use one of the SOLA Custom Analyzers, as detailed below:

- **SOLAAN1:** Will run the user transaction under the SAF ID that is passed using http basic auth.
- **SOLAAN2:** Will run the user transaction under the SAF ID that is passed in the WS-Security Header.
- **XMLPCAN:** Will run the user transaction under the SAF ID that is passed in the WS-Security Header. This is identical to SOLAAN2 except that it is capable of performing identity mapping (if configured).

In order to use any of these special purpose analyzers, you will need to alter your TCPIPS definition in CICS as follows (this can be done by a CICS administrator with CEDA access to the container):

```
ALTER TCPIPS(<name>) G(<groupName>) URM(SOLAAN1/2/XMLPCAN)
```

In order to create a new TCPIPS definition the following command is used:

```
DEFINE TCPIPS(<name>) G(<groupName>)  URM(SOLAAN1/2/XMLPCAN)
PORTNUMBER(nnnn)   PROTOCOL()  -  IIOP, HTTP, ECI
TRANSACTION(CWXN) TSQPREFIX(xxxxxx)
SSL()  AUTHENTICATE()
```

WS Security With SOLA

WS-Security, or Web Services Security, is a standard for the enhancement of SOAP messages to provide security through single message authentication, message integrity and confidentiality.

Tokens

SOLA dictates the following rules for dealing with tokens:

- **UsernameToken:** SOLA supports two kinds of userId tokens:
  - **End Client's UserId:** The current release of SOLA supports mainframe SAF (ACF2/RACF/Top Secret) defined UserId+Password verification. For SOLA to verify a userId/password combination, WS-Policy – done using SOLA IDE's method administrator page -- must be defined to indicate that a UsernameToken is required on input. SOLA uses SAF to verify the password and caches the result. Subsequent verification uses the cache until the cache expires. The cache expiry limit is set during installation default setup (see the section **Installation Default Changes** on page 14). A value of zero for the cache expiry limit will always invoke SAF and no cache will be maintained. Verification using cached data greatly improves performance. Rudimentary testing indicates an eightfold increase in performance overhead with SAF verification compared to cached verification. The cache data is kept in memory in an encrypted form.

    The userId+password combination can be passed in the HTTP Header as basic Auth or according to WS Security specifications.

    The above cache/policy rules will also apply when you pass LDAP UserId+ password.

  - **App Server/Proxy server UserId:** the App/Proxy server UserId is used to identify the requestor and associate a pre-loaded certificate with the request. The public key associated with this certificate is used for Outbound encryption. Certificate pre-loading and association with a userId or token is done using the SOLA IDE. A password is not required for App Server userIds.

    Additional protection can be provided by requiring outbound encryption in the outbound policy. When thus set, the policy requires that the response from SOLA be encrypted using the public key associated with the userId+Certificate combination. Therefore, only the holder of the private key can decrypt it.

- **X509 Token:**X509 Tokens can be used to identify requestors. The requestor sends an X509 Certificate in Base64 format along with signature in the WS Security Header.

  In order for the request to pass security screening, all of the following must be true.

  - RSA Signature Verification must pass.
  - The digest for the specified XML must match with the calculated Hash of canonical form of that XML.
  - The X509 Certificate must be pre-loaded to SOLA.

  There is a performance overhead increase involved for this process. X509 Token verification can't use cache due to the nature of verification and this verification process is repeated for each request.

Encryption

SOLA™

TECH TIP

ICSF must be active on the mainframe for SOLA to perform encryption operations. ICSF, which stands for Integrated Cryptographic Services Facility, is a callable service on the mainframe that provides encryption related functions.

SOLA provides the following encryption features:

- **Inbound:** SOLA provides a self-signed certificate containing a public key that is used for inbound encryption. Currently, SOLA supports the RSA and Triple Des encryption methods. If the input policy does not mandate input encryption, a requestor can send either encrypted or unencrypted SOAP request. SOLA will look at the WS Security Header and will automatically decrypt the data if needed. If the input policy mandates encryption, SOLA will reject unencrypted requests.

- **Outbound:** Outbound encryption requires that SOLA must know which public Key or certificate to use for encrypting the response. In order to do this, the request must identify itself to SOLA. This can be done one of the following two ways.
  - The requestor sends a SOAP request containing an X509 Token and an XML signature. X509 Token verification identifies the requestor and SOLA then uses the public key associate with certificate for outbound encryption.

- The requestor sends a username Token (password is not mandatory). SOLA checks the pre-loaded certificate associated with this username Token and uses the public key associated with the certificate for outbound encryption.

Signature

TECH TIP

ICSF must be active on the mainframe for SOLA to perform signature creation and verification. ICSF, which stands for Integrated Cryptographic Services Facility, is a callable service on the mainframe that provides encryption related functions.

SOLA provides the following signature verification features:

- **Inbound:** this is the same as the X509 Token verification process discussed earlier. Although the policy may or many not require signatures, SOLA can automatically detect and verify signatures, if present.
- **Outbound:** not supported in the current release.

Combinations

All of SOLA's WS Security features can be used in combination with one another for an even higher degree of security.

- **Username Token + Outbound Encryption:** this combination has been discussed under outbound encryption.

- **X509Token + Outbound Encryption:** this has been discussed in section 2 under outbound encryption.

- **Inbound (Encryption + Signature):** SOLA requires that the requestor must first encrypt and then sign an inbound request. SOLA currently does not support the verification of requests where a signature is created before encryption is performed.

Channel and Resource Lockdown

Although SOLA implements WS-Security, it also ships with a built-in custom security policy that implements Channel and Resource Lockdown. For any given operation/method you may choose to use this custom security or any other supported WS-Security mechanism. Note that all security policies (WS-Security or SOLA's Channel and Resource Lockdown security) are enforced at runtime by SOLA's security exit program XMLPC080.

- **Channel Lockdown:** at this level, SOLA checks to insure that the client's IP address is registered (trusted) with SOLA.
- **Resource Lockdown:** at this level, SOLA determines if access to the enterprise resource (i.e. the program) that hosts the requested Web Service is allowed from the client's IP address.

All of SOLA's custom security information is stored and maintained in a series of DB2 tables. For performance reasons, this data is held in memory to avoid I/O operations during the runtime process.

At runtime, if a SOLA request does not find the security data in memory, it will spawn off a process that will read the security data from DB2 and build the appropriate storage queues to contain the security data. All subsequent SOLA requests will read the security matrix directly from core storage.

Typically, these queues will be created shortly after startup. However, there are SOLA administration features that will rebuild some or all of these queues on request. The ability to implement changes on request is important, since any changes made to the DB2 tables after startup will not take effect until the SOLA security queues are rebuilt.

Policies

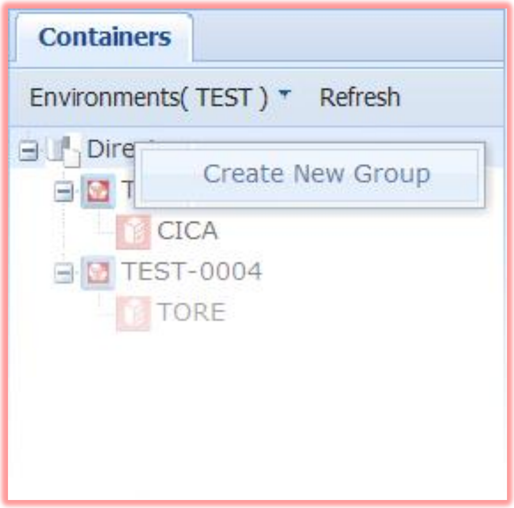
There are two types of policies in SOLA, the default policy and the method-specific policy. Both the default and method-specific policies define settings that pertain to security, while only the method-specific policy defines settings that pertain to auditing.

If a method-specific policy exists, it will always override the default policy. The default policy, which can be enabled or disabled, comes into effect when a method does not have its own policy (and the default policy is enabled).

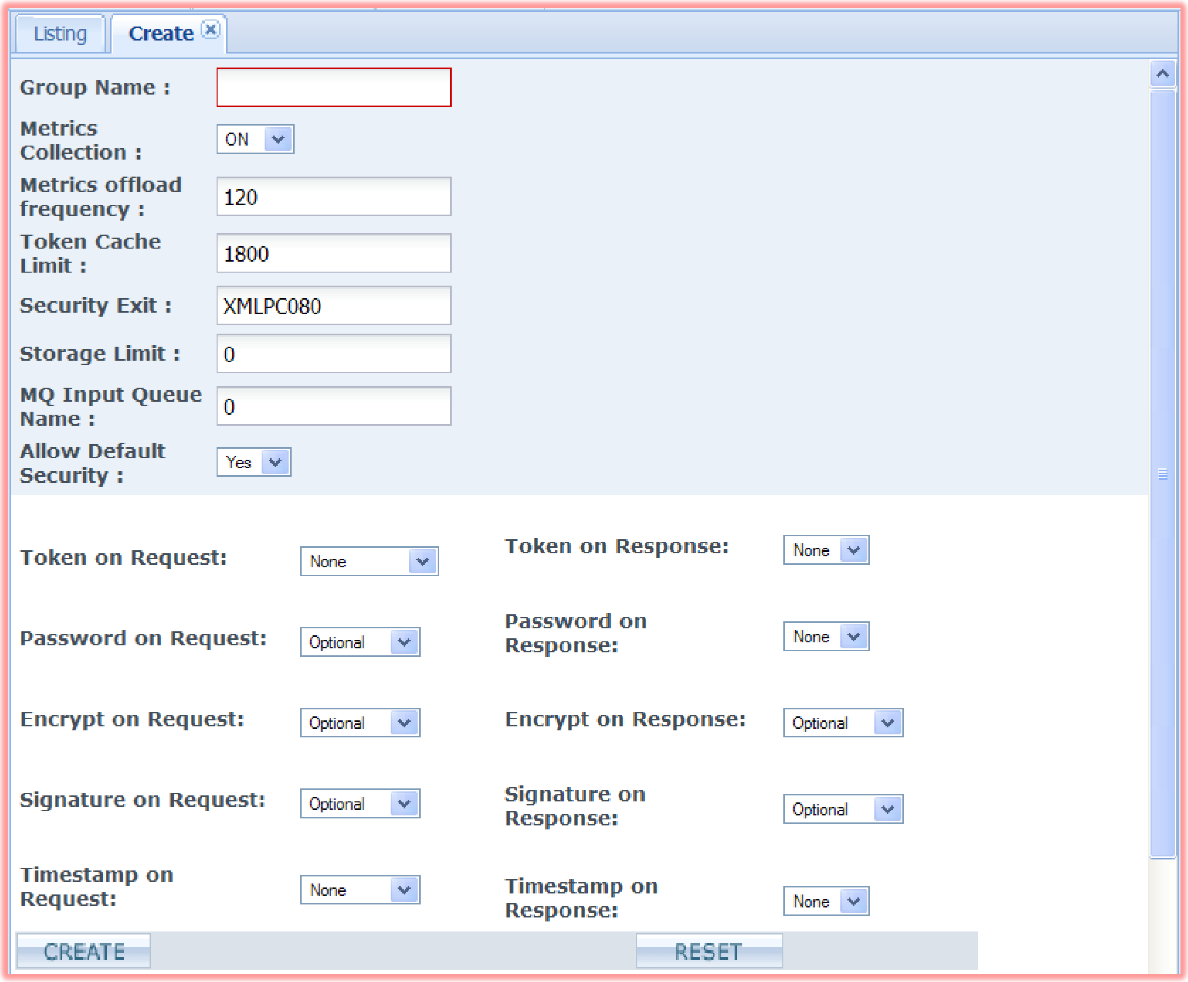
Default Security Policy

A default policy exists per *Container Group*. You can specify a default policy when you create a Container Group. Once specified, the policy settings become properties of the Container Group, and can be modified through the property pane.

Policies are managed through SOLA's Resource Manager and in SOLA Developer (at the Project/Program/Method level in the Directory via the Program Menu option – Policy Management.) You create a default policy for a Container Group when you create that group. Right click on the directory icon in the containers list and choose Create New Group.



The following screen will be displayed:



Group Name :	
Metrics Collection :	ON
Metrics offload frequency :	120
Token Cache Limit :	1800
Security Exit :	XMLPC080
Storage Limit :	0
MQ Input Queue Name :	0
Allow Default Security :	Yes

Token on Request:	None	Token on Response:	None
Password on Request:	Optional	Password on Response:	None
Encrypt on Request:	Optional	Encrypt on Response:	Optional
Signature on Request:	Optional	Signature on Response:	Optional
Timestamp on Request:	None	Timestamp on Response:	None

Figure 1 – Default Security Policy

This screen will allow you to view and make changes to the default security policy. There are two sets of settings, one for inbound (requests) and one for outbound (responses).

Security Policy Settings:

- **Is Default Security Policy Allowed:** If a specific security policy is not defined for a transaction and this setting is set to YES, SOLA can use the installation default policy, defined in this screen. If this setting is set to NO, all transactions without a defined policy will be rejected.

- **Inbound/Outbound Security Token:** this setting determines whether SOLA will accept requests without an attached security token and whether SOLA will attach a security token to responses. An inbound setting of MF User Id, LDAP Id or Custom means that SOLA will reject requests without the specified type of attached tokens and an outbound setting of YES means that SOLA will attach tokens to all responses. A setting of NO will allow requests without security tokens (inbound) and will not attach tokens to responses(outbound).

- **Inbound/Outbound Password:** this setting is only used if the Security Token setting is set to a value other than NO. The setting determines whether SOLA will accept requests without an attached password (inbound setting) and whether SOLA will attach a password to responses (outbound setting). A setting NO will allow requests without attached passwords and will not attach passwords to responses.
- **Inbound/Outbound Encryption:** this setting determines whether SOLA accepts requests that are not encrypted and whether SOLA will encrypt responses.
- **Inbound/Outbound Signature:** this setting determines whether SOLA accepts requests without an attached signature and whether SOLA will attach a signature to responses.

Click **Create** to save changes.

Creating and Managing Policies

Creating policies works a bit differently than creating other subjects such as users or IP addresses. For all other subjects, the individual tree item is the subject, and the group is a container for the subjects. With policies, the group is the policy, and the items in a group are aspects of that policy.

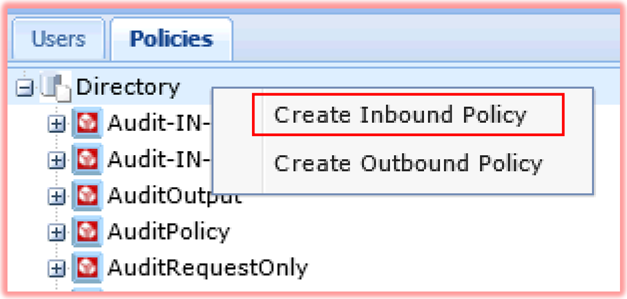
There are no permissible drag and drop operations within the policy tree itself, the only allowable drag and dropping of policies is to assign them to a resource.

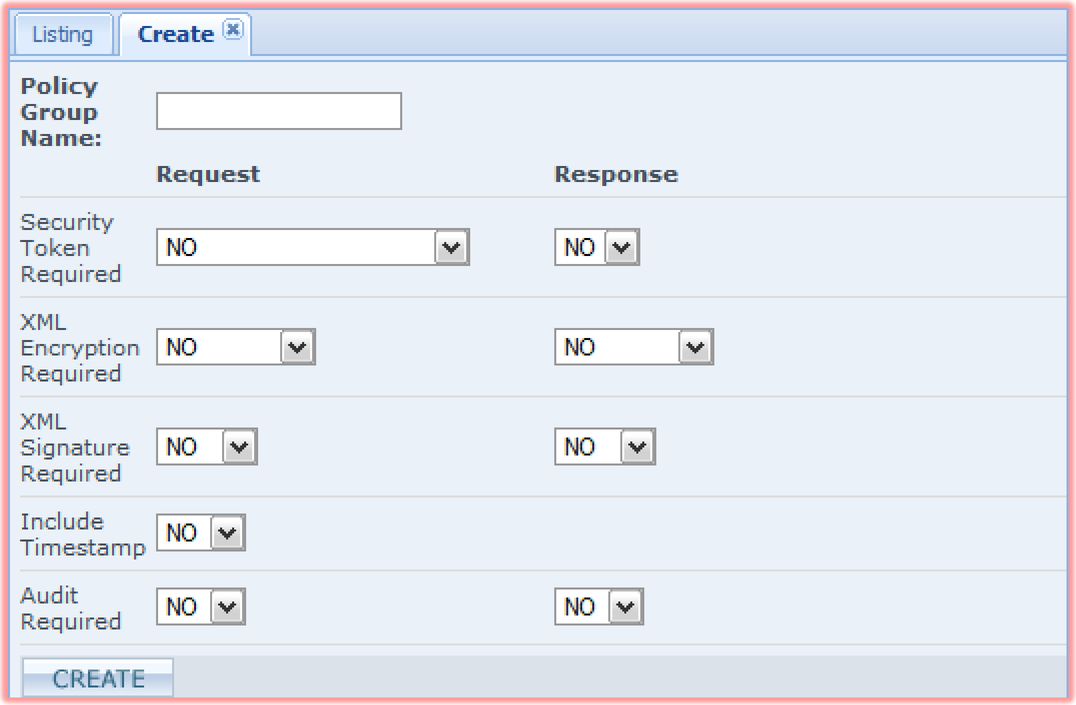
There are 2 types of policies that can be created.

- Inbound Policy to be applied on SOLA inbound services/operations
- Outbound Policy to be applied on SOLA outbound services/operations

Creating and Managing Inbound Policies

To create a new inbound policy, right click the directory root and select **Create Inbound Policy** from the menu. This will display the Create tab, allowing you to create a new policy.





Policy Group Name:

	Request	Response
Security Token Required	NO	NO
XML Encryption Required	NO	NO
XML Signature Required	NO	NO
Include Timestamp	NO	
Audit Required	NO	NO

Enter a policy name in the Policy Group Name field, then configure the policy using the options shown below.

Input (Request) Settings

Security Token Required: this setting determines whether SOLA will accept requests without an attached security token. The type of token required can be defined in a container group policy.

- **NO:** SOLA will allow requests without security tokens
- **Username Token:** SOLA will only accept requests with username tokens.
- **Encrypted Username Token:** SOLA will only accept requests with encrypted username tokens.
- **SAML Token:** SOLA will require SAML credentials as a security token.

XML Encryption Required: this setting determines whether SOLA accepts requests that are not encrypted.

- **NO:** encryption is not required (SOLA will accept requests without encryption).
- **RSA-3DES:** encryption is required, and must be RSA-3DES (more schema options will be available with future versions of SOLA).

XML Signature Required: this setting determines whether SOLA accepts requests without an attached signature.

- **NO:** attached signatures are not required (SOLA will accept requests without attached signatures).
- **Body:** the body of the SOAP request must be signed.

Include Timestamp: this setting determines whether SOLA accepts requests without an attached timestamp. The timestamp contains the policy's expiration date and time.

- **NO:** attached timestamps are not required (SOLA will accept requests without attached timestamps).
- **YES:** attached timestamps are required.

Audit Required: this setting determines whether SOLA will trace input XML, thereby auditing request.

- **NO:** instructs SOLA not to audit input XML.
- **YES:** instructs SOLA to trace the input XML, thereby auditing requests.

Output (Response) Settings

Security Token Required: this setting determines whether SOLA will attach a security token to responses. With the current version of SOLA, the only option is NO.

XML Encryption Required: this setting determines whether SOLA will encrypt responses.

- **NO:** encryption is not required (SOLA will not encrypt responses).
- **RSA-3DES:** encryption is required, and SOLA will use RSA-3DES (more schema options will be available with future versions of SOLA).

XML Signature Required: this setting determines whether SOLA will attach a signature to responses.

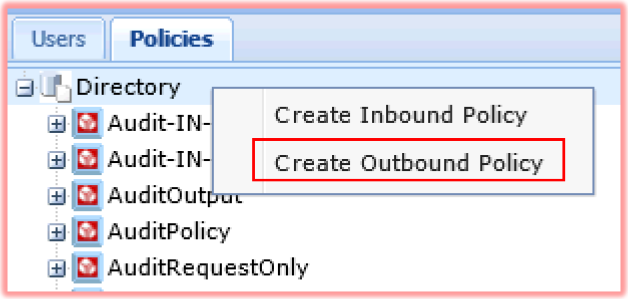
- **NO:** SOLA will not attach signatures to responses.
- **Body:** SOLA will attach a signature to the body of responses.

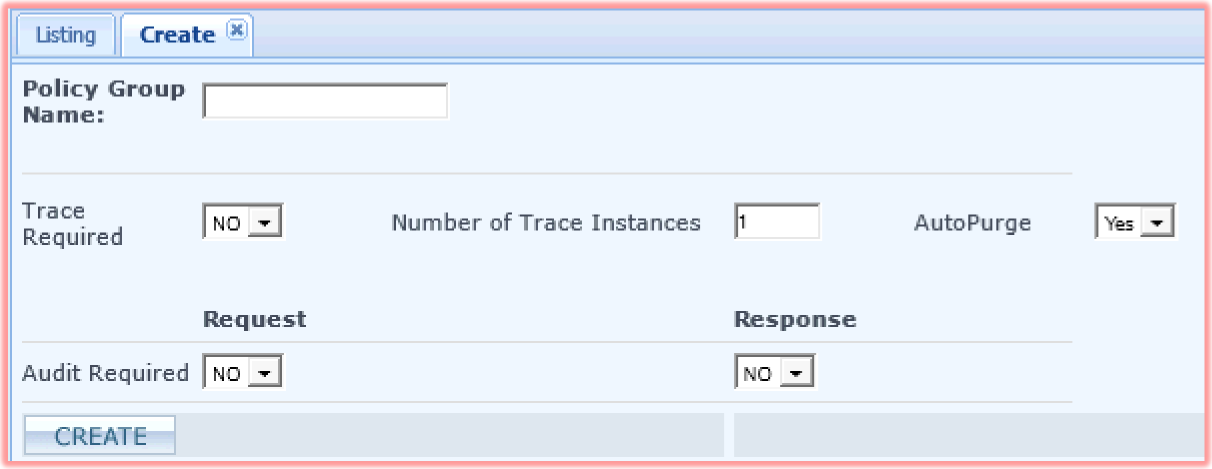
Audit Required: this setting determines whether SOLA will trace output XML, thereby auditing responses.

- **NO:** instructs SOLA not to trace output XML.
- **YES:** instructs SOLA to trace output XML, thereby auditing responses.

Creating and Managing Outbound Policies

To create a new outbound policy, right click the directory root and select **Create Outbound Policy** from the menu. This will display the Create tab, allowing you to create a new policy.





Policy Group Name:					
Trace Required	NO	Number of Trace Instances	1	AutoPurge	Yes
	Request		Response		
Audit Required	NO		NO		

Enter a policy name in the Policy Group Name field, then configure the policy using the options shown below.

Trace Required: this setting determines whether SOLA Outbound Tracing is Required or not.

- **NO:** SOLA will not generate outbound trace
- **Yes:** SOLA will generate outbound trace. In CICS, SOLA trace will be generated in a TSQ having naming convention ST-<ProgramName>-nnnn
  Where <ProgramName> is the directory name of the Outbound program captured in SOLA and nnnn is the running sequence number of the trace

Number of Trace Instances : this setting indicates the number of trace instances to be captured. Maximum number of trace instances that can be setup is 9999.

AutoPurge: this setting determines whether trace entries captured by SOLA needs to be auto purged

- **NO:** SOLA will not Autopurge trace entries. If tracing is already captured for maximum number of trace instances as setup in the policy then further tracing will be disabled
- **Yes:** SOLA will Autopurge trace entries. If tracing is already captured for maximum number of trace instances as setup in the policy then oldest trace instance is purged and new trace instance is generated.

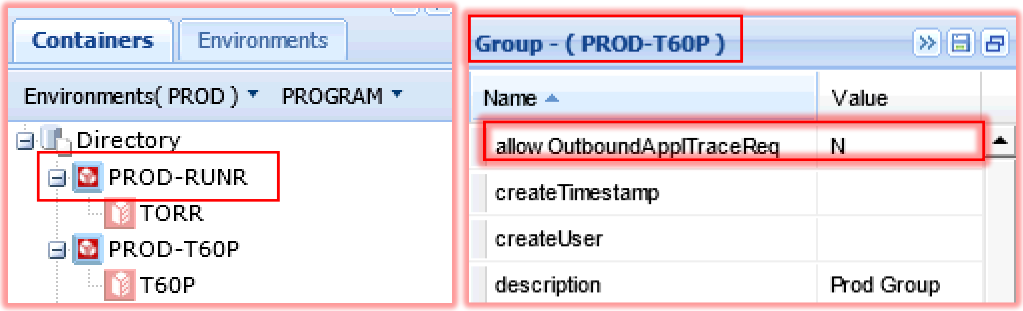
Audit Required: this setting determines whether SOLA will audit SOAP request and/or response XML.

- **NO:** instructs SOLA not to audit
- **YES:** instructs SOLA to audit

Note: *Outbound Policy Support is currently only available for applications that invoke SOLA outbound plugin under CICS.*

Managing Application enabled SOLA Outbound Tracing

SOLA outbound plugin supports application enabled tracing where by application sets WSC-INVOKE-TRACE field in the generated interface copybook area to 'Y'. This enables tracing and trace under CICS is generated under SOLATRACE TSQ. To disable application enabled tracing on a container, select the container group and update the property "allowOutboundApplTraceReq" to "N" and save the changes as shown in the following illustration. This action will disable any application enabled tracing. Tracing is only enabled with the assignment of outbound tracing policy.



Group - (PROD-T60P)

Name	Value
allowOutboundApplTraceReq	N
createTimestamp	
createUser	
description	Prod Group

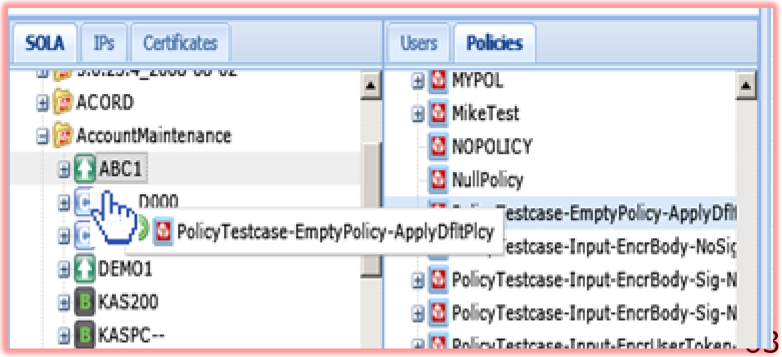
Assigning a Policy

TECH TIP

SOLA Developer Studio has a new Policy Management function enabled as a sub-menu under the context of a Program. This interface is a more intuitive interface for managing assignment/deployment of policies

Once you have a policy, you can assign the policy to a resource or group of resources in the resource panel. This assignment is accomplished using Resource Manager's drag and drop capabilities; a subject is dragged over to a resource and dropped into it.

In the following illustration, a policy is being dragged to program ABC1. The policy will be assigned to the program and all of its methods.


SOLA
IPs
Certificates
Users
Policies
ACORD
AccountMaintenance
ABC1
D000
PolicyTestcase-EmptyPolicy-ApplyDfltPlcy
DEMO1
KAS200
KASPC--
MYPOL
MikeTest
NOPOLICY
NullPolicy
estcase-EmptyPolicy-ApplyDflt
estcase-Input-EncrBody-NoSig
PolicyTestcase-Input-EncrBody-Sig-N
PolicyTestcase-Input-EncrBody-Sig-N

The following table illustrates the effects of various associations:

Subject	Resource	Result
Policy	Program	The policy will be assigned to the program and all of its methods. Once deployed, the program will use this policy, overriding the container default policy, except where the default policy defines a requirement set by the assigned policy.
Policy	Method	The policy will be assigned to the method only. Once deployed, the method will use this policy, overriding the container default policy, except where the default policy defines a requirement set by the assigned policy. If a policy is applied at both Program and Method level then Method level policy overrides the Program level policy

Once a policy is assigned to a resource, you can view it by right clicking on the resource and selecting **Show Policies**.

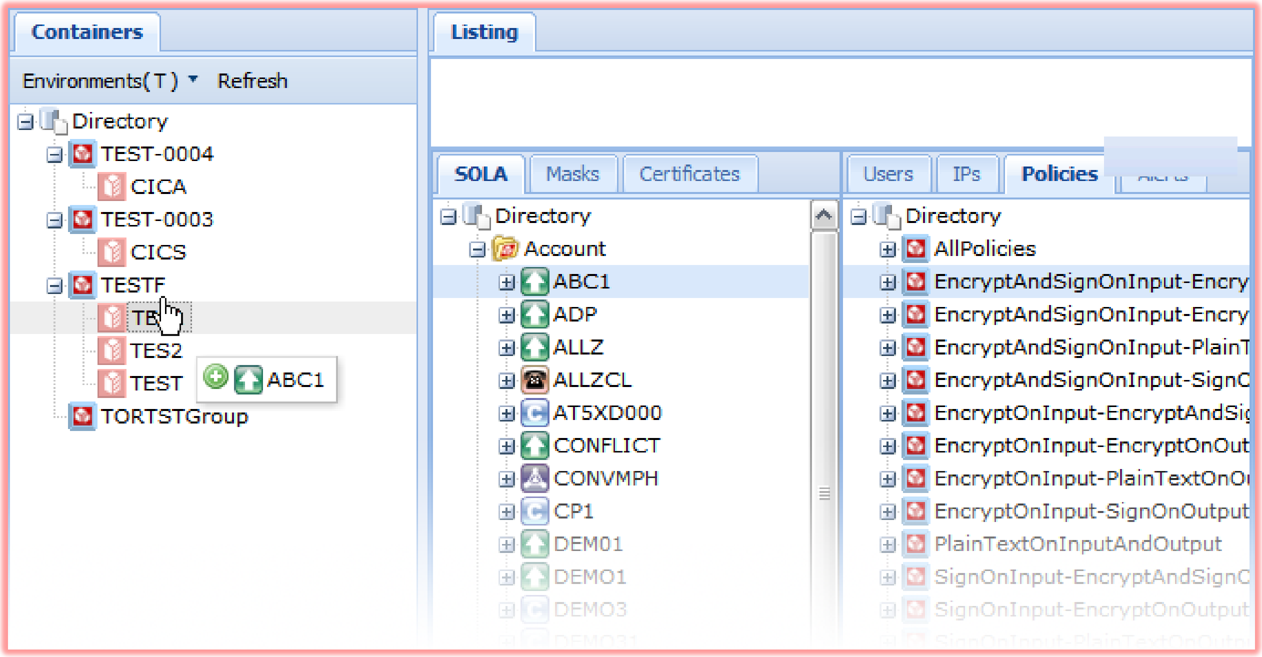
Deploying a Policy

When a policy is assigned to a resource, it is not active on a runtime container until it is deployed. For example, if you drag a policy from the subjects panel to a program in the resource panel, the policy is assigned to that resource, but is not in effect. To put it into effect, you must deploy that assigned policy into a target SOLA runtime container group.

The deployment of an assigned policy is accomplished using Resource Manager's drag and drop capabilities; an assigned policy is dragged over to a SOLA container, activating that assigned policy for every container in the target container's group.

NOTE: although Resource Manager does not allow subjects to be dragged into container groups, a subject can only be activated for a container group, not individual containers. Dragging an assigned policy into a container deploys that assigned policy to every container in that container's parent group.

The following illustration shows a policy being deployed in the TESTF container group. It could have been dragged to any container in the TESTF group with the same results.


Containers
Environments(T) ▾ Refresh
Directory
TEST-0004
CICA
TEST-0003
CICS
TESTF
TE
TES2
TEST
ABC1
TORTSTGroup
Listing
SOLA
Masks
Certificates
Directory
Account
ABC1
ADP
ALLZ
ALLZCL
AT5XD000
CONFLICT
CONVMPH
CP1
DEM01
DEMO1
DEMO3
Users
IPs
Policies
Directory
AllPolicies
EncryptAndSignOnInput-Encry
EncryptAndSignOnInput-Encry
EncryptAndSignOnInput-PlainT
EncryptAndSignOnInput-SignC
EncryptOnInput-EncryptAndSi
EncryptOnInput-EncryptOnOut
EncryptOnInput-PlainTextOnO
EncryptOnInput-SignOnOutput
PlainTextOnInputAndOutput
SignOnInput-EncryptAndSignC
SignOnInput-EncryptOnOutput

Spooling Trace TSQ to JES

The following is the process for spooling SOLA Traces captured in CICS TSQs. Add following special endpoint to endpoints.xml file using Admin screens (Refer Page 47).

http://<MainframeHost>:<CICSPort>/CICS/XML3/XMLPC003?Request=spool

<CICSPort> corresponds to the TCPIP listener port on the region where the trace has been generated.

Click on 'SOAP Test' icon on SOLA developer studio to open Raw Soap Test tab. Select the above endpoint and type in the request area

<TSQ>TraceTSQName</TSQ>

To spool SOLATRACE TSQ detail enter the request as <TSQ>SOLATRACE</TSQ>

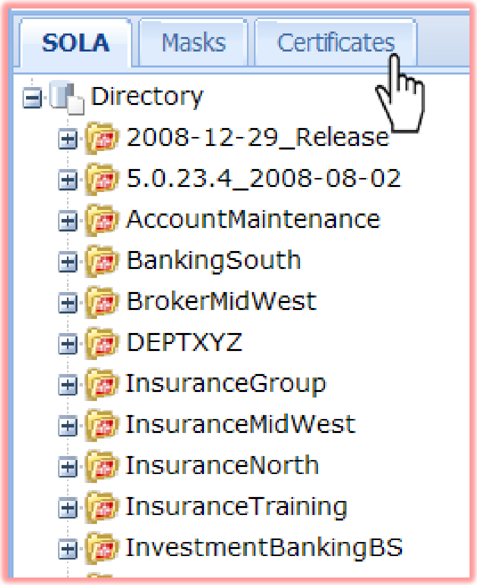
To use this feature you must enable **SPOOL=YES** in the regions SIT Override.

The spooled traces can be looked under the active Joblog of the CICS region and the spool files have a DDName naming convention as Snnnnnnn

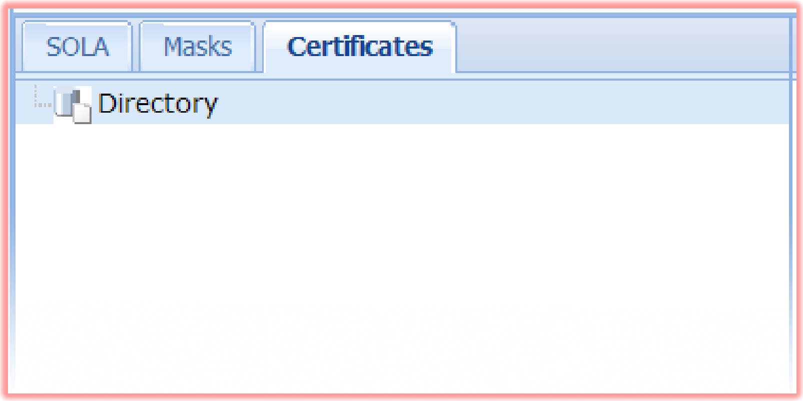
Certificates

SOLA uses digital certificates to encrypt and decrypt XML, and to sign and validate digital signatures. SOLA supports a Public Key/Private Key infrastructure, where messages that are encrypted with one of the pair of certificates can only be decrypted with the matching certificate from the pair. SOLA ships with a single Public Key/Private Key pair, but it is able to support multiple Public Key/Private Key pairs by using ICSF as a keystore.

Downloading and uploading certificates in SOLA is done using the SOLA Resource Manager. Click on the Certificates tab to access

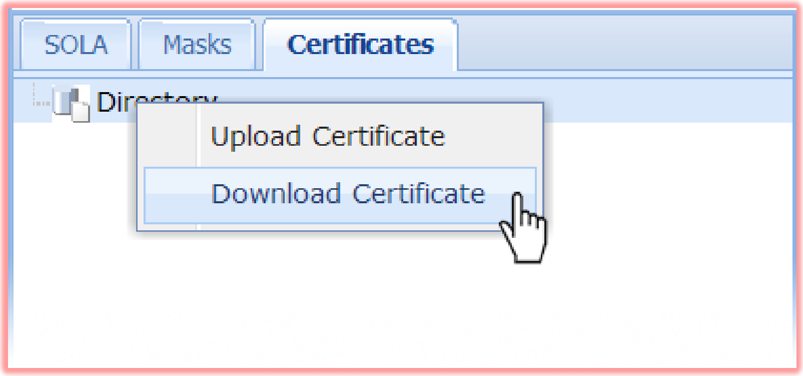


The following screen will be displayed:

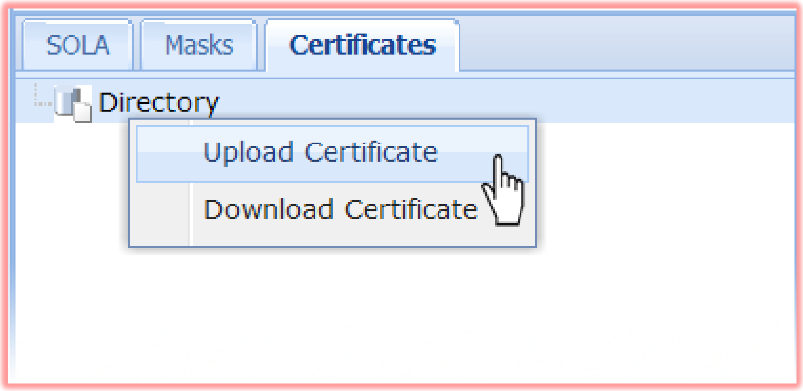


Download SOLA's Certificate

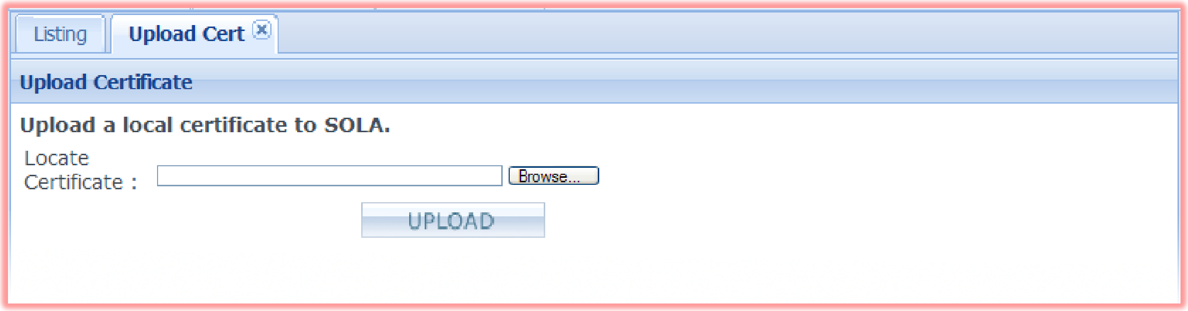
To send any sort of encrypted file to SOLA requires the use of SOLA's X509 certificate, which you will need to download. Right clicking the SOLA Certificate and selecting **Download SOLA Certificate** will allow you to download this certificate to your local machine.



Upload a Certificate to SOLA



Right click on the directory icon and select **Upload Certificate** to display the upload X509 Certificate screen.



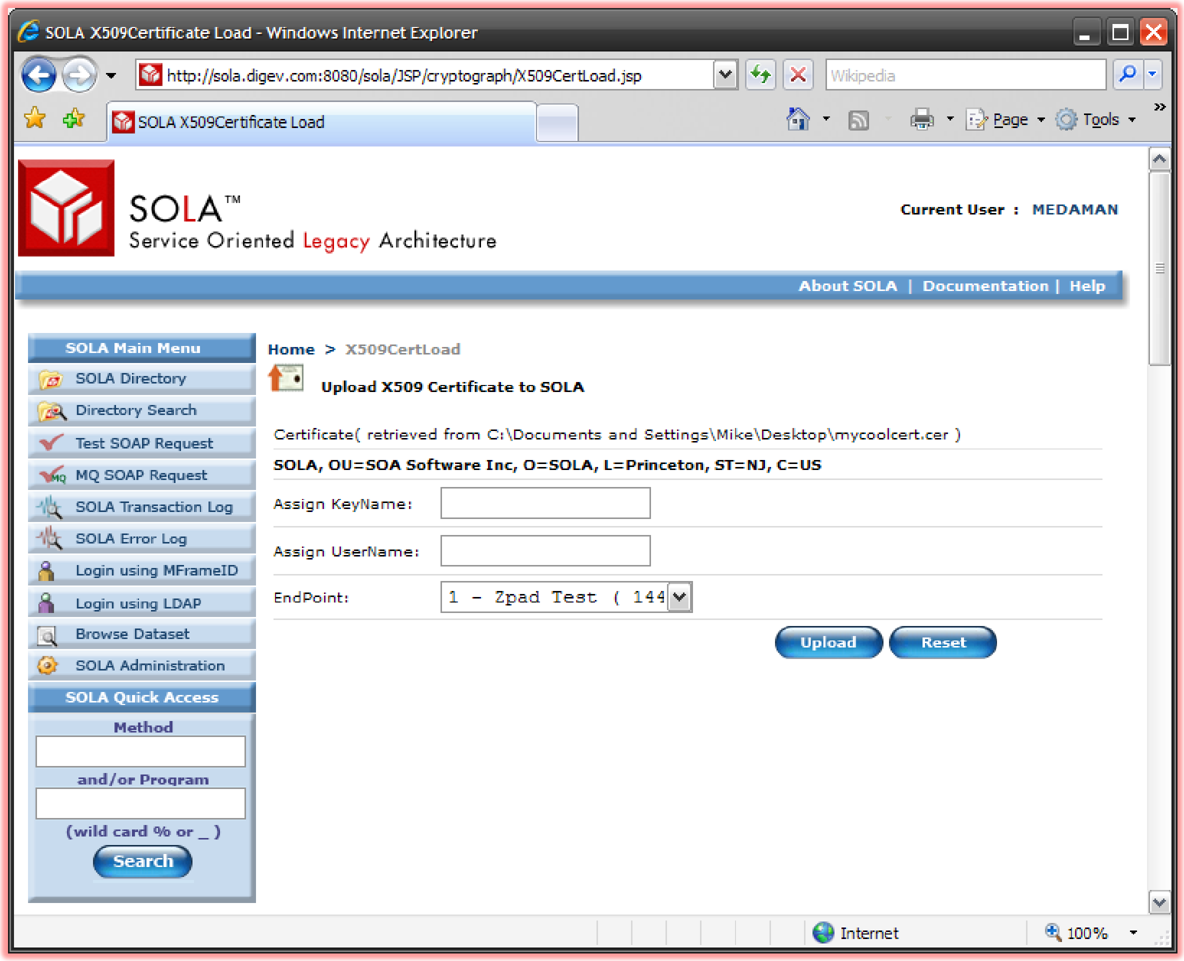
Upload a local certificate to SOLA.

Locate Certificate :

Click **Browse** to locate the certificate, then click **Retrieve** to upload the certificate to the J2EE server that hosts SOLA's IDE (at this point the certificate is not yet uploaded to the mainframe – it is uploaded to the mainframe in the next step).

This screen allows you to pre-load an X509 certificate to SOLA.

Doing so will display the following screen.



Upload X509 Certificate to SOLA

Certificate(retrieved from C:\Documents and Settings\Mike\Desktop\mycoolcert.cer)

SOLA, OU=SOA Software Inc, O=SOLA, L=Princeton, ST=NJ, C=US

Assign KeyName:

Assign UserName:

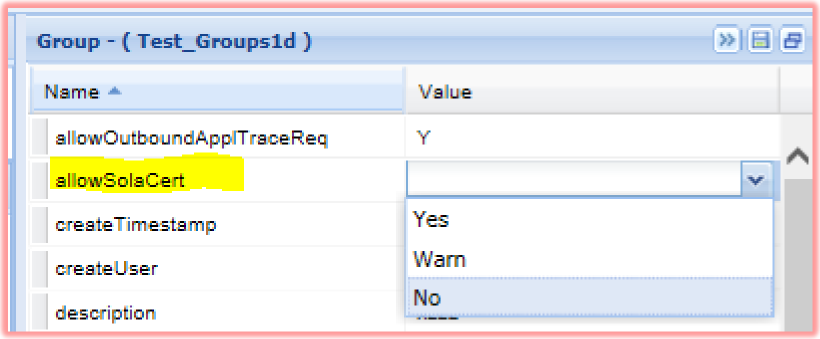
EndPoint: 1 – Zpad Test (144

Using the **Assign KeyName** and **Assign UserName** fields, you can assign a key name and/or a username to the certificate. A key name can be used instead of a certificate once the certificate is uploaded to SOLA, while a username associates a mainframe username with the certificate, enabling the certificate to be used to access mainframe functions.

Click **Upload** when finished, or **Reset** to undo your changes.

- An attribute "allowSOLACert" is captured at the container level. If the attribute is not set or not available then the default behavior is to support use of SOLA product certificate for processing encryption/decryption/signature. The attribute if captured can be set to "Yes", "Warn", "No".

1. Option '**Yes**' is same as current default to allow usage of SOLA Certificate.
2. Option '**Warn**' when set will allow applications to use SOLA certificate but warnings are generated in SOLA Logs. This option can help to identify applications that use SOLA default certificate.
3. Option '**No**' when set will reject the request if it uses SOLA certificate.



Group - (Test_Groups1d)

Name	Value
allowOutboundApplTraceReq	Y
allowSolaCert	
createTimestamp	
createUser	
description	

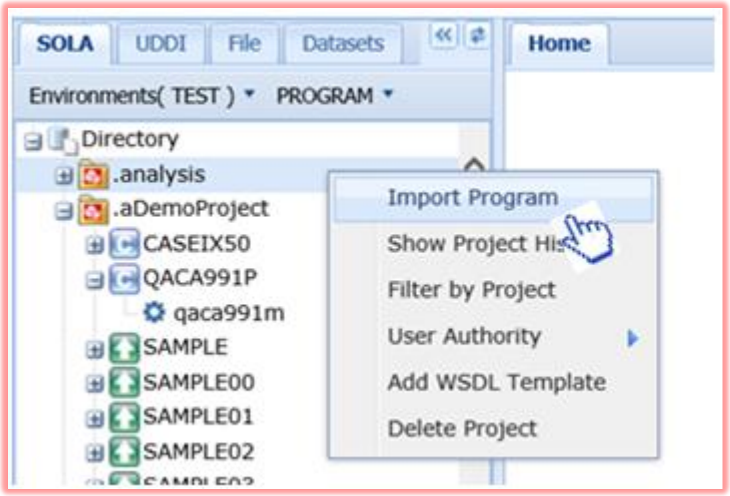
Project Administration

There are multiple ways to administer a Project. You will need Project Administrator rights to perform these tasks.

Note: Prior releases of SOLA allowed you to customize JCL at the project level. With the customizable schema facility introduced in SOLA 6.0, this capability is no longer necessary. JCL is now only modifiable at the SOLA installation level.

All project administration tasks are performed through the project menu. Click on any project to access the project menu for that project.

Project Menu



Import Program: displays the import panel, which can be used to import legacy programs and create web services. **The import panel is the gateway to creating web services from every program type that SOLA supports.** This function is described in detail in the SOLA Users Guide with descriptions of the import and analysis process for every supported program. Refer to the SOLA Users Guide to get information about the program type you are interested in.

Show Project History: displays a list of changes to the project from the day it was created, with date and time stamps for each change. See Figure 1 below:

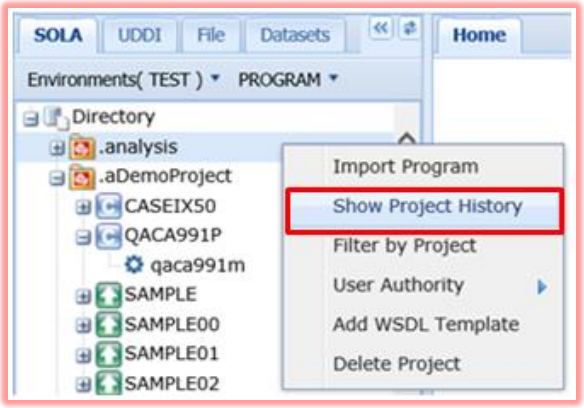
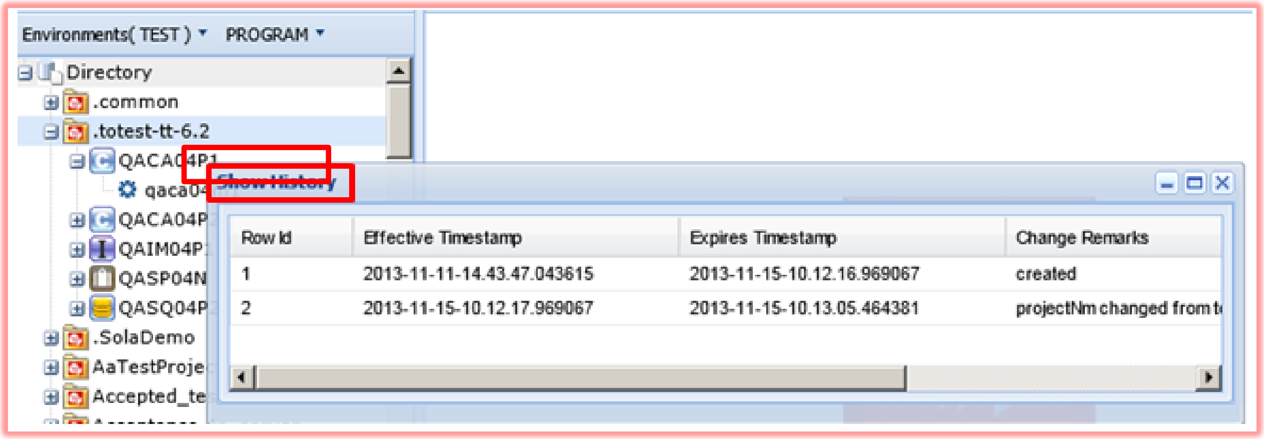
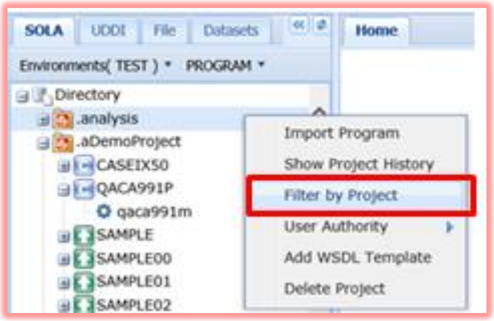


Figure 1:



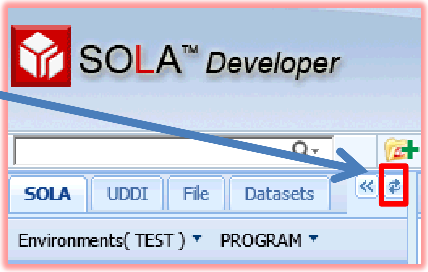
Row Id	Effective Timestamp	Expires Timestamp	Change Remarks
1	2013-11-11-14.43.47.043615	2013-11-15-10.12.16.969067	created
2	2013-11-15-10.12.17.969067	2013-11-15-10.13.05.464381	projectNm changed from t

Filter by Project: enables a user to view only a specific project and the programs or services within the project, in the IDE directory tree. A user can share a Project specific URL within a team so developers only see objects within the project when using that URL. An example of this would be:

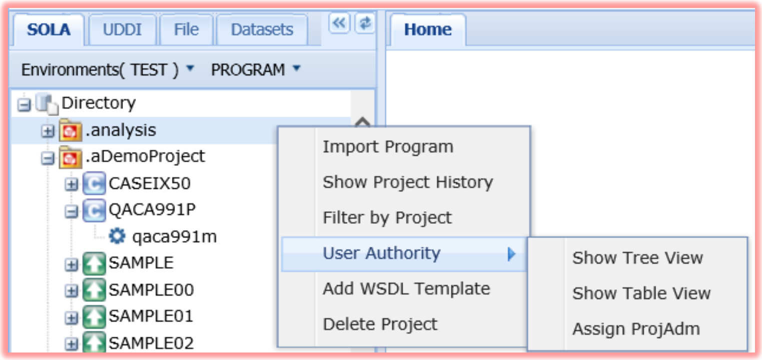


http://<IDEHost>:<IDEPort>/sola/index.html?project=<ProjectName>

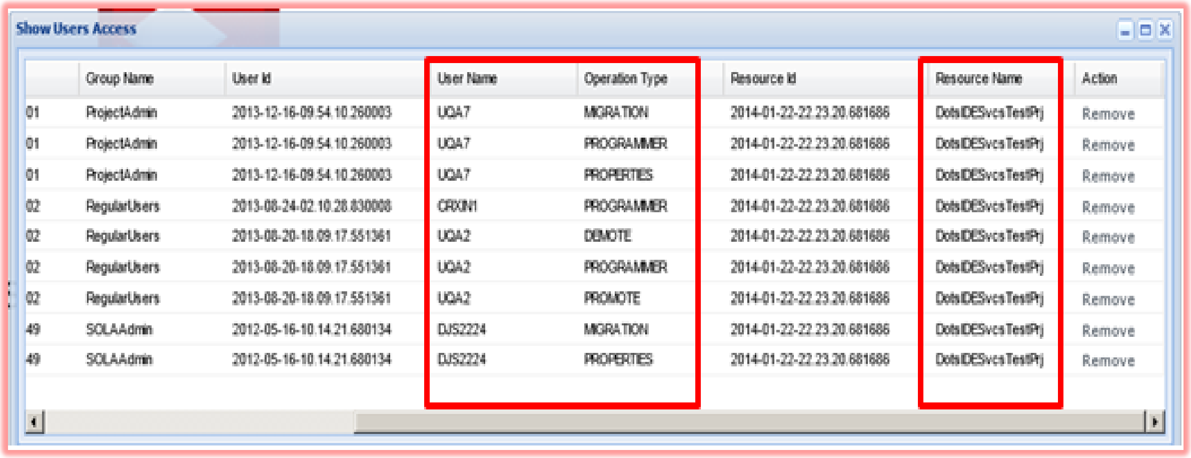
Note: To clear the filtering criteria click on the refresh button (⟳) as shown below:



User Authority: Enables the management and assignment of the SOLA User's access. Clicking on **User Authority** will display the panel **User Authority Manager** after clicking on the **Show Tree View** option. User Authority functionality is restricted and must be granted by the SOLA Administrator or Project Administrator. For more detailed information about SOLA Developer – User Authority see page 31 in the SOLA Users Guide.



Show Table View: clicking on Show Table View for the Project displays a list of the access **Operation Type** each **User Name** has been assigned for the selected Project/**Resource Name** as seen in the illustration below:



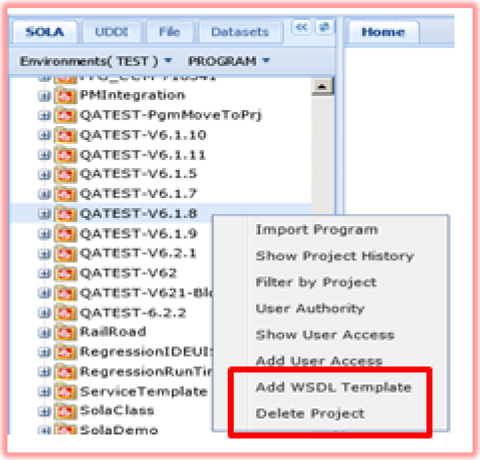
Show Users Access

	Group Name	User Id	User Name	Operation Type	Resource Id	Resource Name	Action
01	ProjectAdmin	2013-12-16-09.54.10.260003	UQA7	MIGRATION	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
01	ProjectAdmin	2013-12-16-09.54.10.260003	UQA7	PROGRAMMER	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
01	ProjectAdmin	2013-12-16-09.54.10.260003	UQA7	PROPERTIES	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
02	RegularUsers	2013-08-24-02.10.28.830008	CRXIN1	PROGRAMMER	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
02	RegularUsers	2013-08-20-18.09.17.551361	UQA2	DEMOTE	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
02	RegularUsers	2013-08-20-18.09.17.551361	UQA2	PROGRAMMER	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
02	RegularUsers	2013-08-20-18.09.17.551361	UQA2	PROMOTE	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
49	SOLAAdmin	2012-05-16-10.14.21.680134	DJS2224	MIGRATION	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove
49	SOLAAdmin	2012-05-16-10.14.21.680134	DJS2224	PROPERTIES	2014-01-22-22.23.20.681686	DotsIDESvcsTestPrj	Remove

Assign ProjAdm: authorizes a user to work on the project as a Project Administrator. Only the SOLA Administrator can grant this access.

Add WSDL Template: allows users to add a WSDL template containing data defining tags that will be included with all requests and responses sent by programs within the project. This is used to add data tags that are not included when importing a legacy program.

Delete Project: deletes the selected project and all programs and methods in the project. There is no undo function.



Add/Update JCL

There are two points in the life-cycle of a service where SOLA will submit an z/OS batch job. You can customize the JCL for those jobs to allow SOLA to integrate with your environment's change management system.

The two points are:

- When you click finalize during Analysis
- When you Promote or Demote a program

The customization points allow you to:

- Store a template in your change management system
- Synchronize the promotion/demotion of a program with your change management system

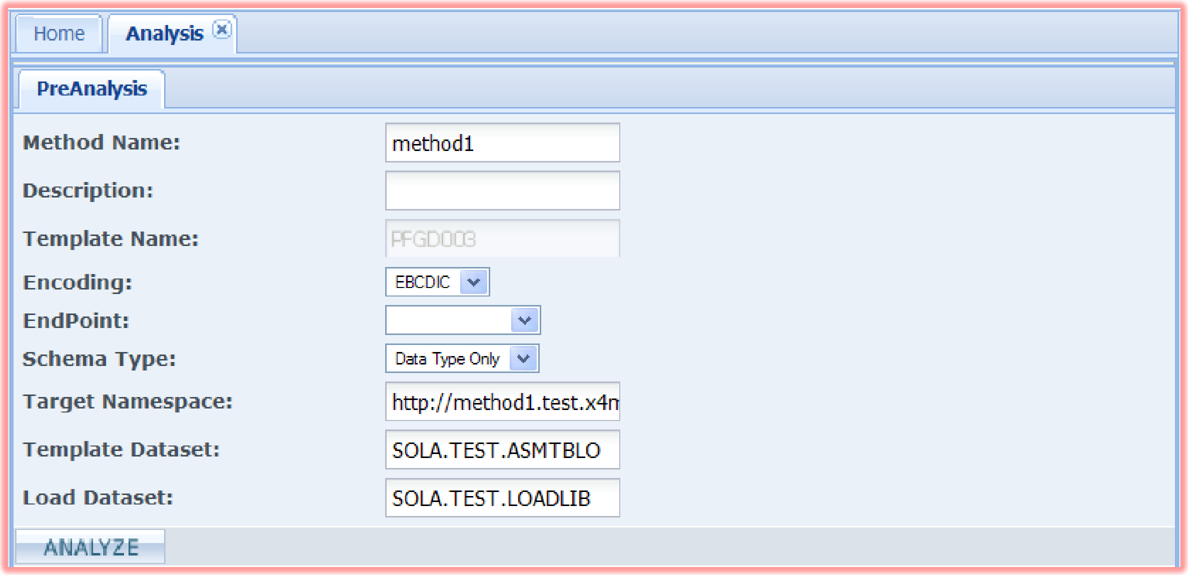
The JCL for those two jobs can be modified by using the SOLA File Editor, which can be found under the Admin menu.

Before we begin, a simple explanation of the way that SOLA interacts with the z/OS side of your mainframe will help you understand what's required.

The SOLA Server (which runs the SOLA Developer J2EE application) communicates with the z/OS mainframe in two ways:

1. Using SOLA Web Services to retrieve and store data from SOLA's Directory (which is a mainframe DB2 database). These web services interact with the SOLA container that's designated by the property **SOLASoapAddress** in the /inst /system /endpoints.xml property file.
2. Using the z/OS Communications Server to retrieve and store sequential files, PDS members and JES spoolfiles.

When you are doing Analysis to create/modify a web service, the SOLA Developer asks you for two dataset names, as shown below:



Method Name:	method1
Description:	
Template Name:	PFGD003
Encoding:	EBCDIC
EndPoint:	
Schema Type:	Data Type Only
Target Namespace:	http://method1.test.x4n
Template Dataset:	SOLA.TEST.ASMTBLO
Load Dataset:	SOLA.TEST.LOADLIB

The datasets are **Template Dataset** and **Load Dataset**. **Template Dataset** names a PDS file (LRECL=80) where the source of the assembler metadata template will be stored. SOLA will assemble and link-edit the template, storing the load module in **Load Dataset**, which is a Load Library (concatenated to DFHRPL if your SOLA Container runs in CICS). SOLA uses the JCL from Jobcard.txt and Assembler.txt to create the job that it submits. Both of these files can be found in /inst /system.

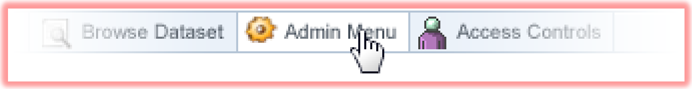
Template Dataset and Load Dataset can be entered on the PreAnalysis screen, or can be saved in your user profile.

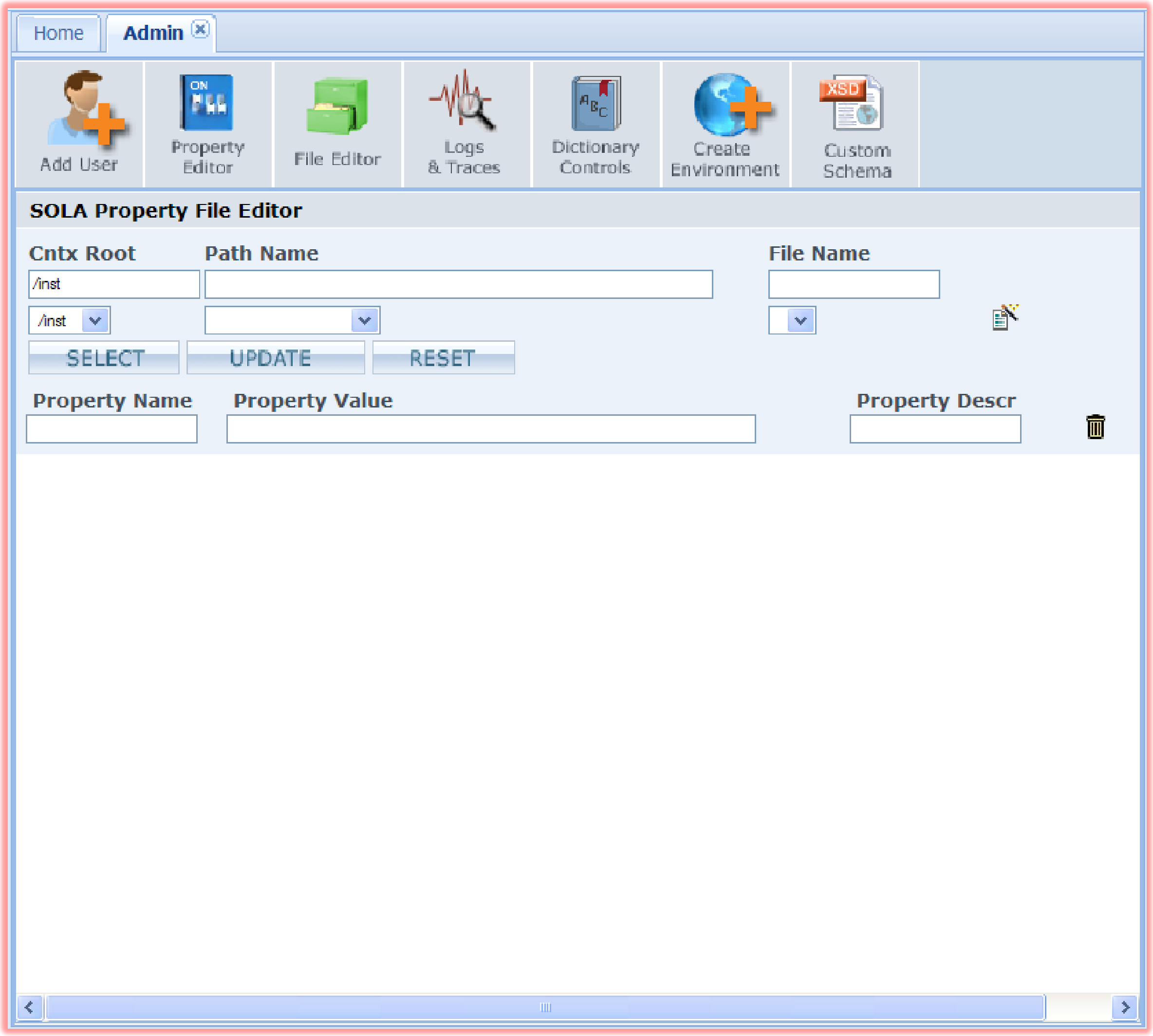
When you click the **Finalize** button the sequence of operations is:

1. SOLA stores the service in the SOLA Directory
2. SOLA generates the WSDL, which it stores on the file system of the SOLA Server
3. SOLA generates the template as an assembler source file, and it stores that source as a member of the PDS named in **Template Dataset**.
4. SOLA generates a job to assemble and link-edit the template,and it stores that job in a mainframe sequential file <userid>.MVSJCL.
5. SOLA submits job <userid>.MVSJCL
6. When the job has completed successfully, SOLA issues a NEWCOPY (CICS only) in the endpoint region.

You can customize the Jobcard.txt and Assembler.txt files by using the SOLA Property File editor.

To make changes to SOLA system properties, access the SOLA Admin Menu screen by selecting **SOLA Administration** from the SOLA Home page.





SOLA Property File Editor

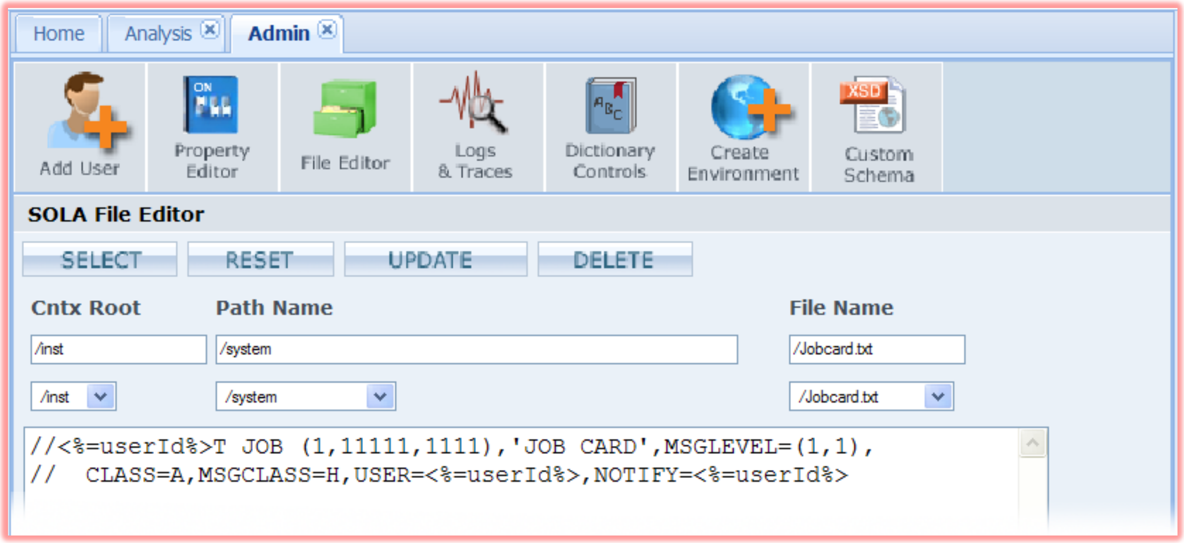
Cntx Root: /inst
Path Name:
File Name:

Cntx Root (drop-down): /inst

Property Name:
Property Value:
Property Descr:

Click the **File Editor** icon to display the File Editor screen set to File Editor Mode.

To edit the Jobcard.txt file, you would enter /inst /system and /Jobcard.txt.

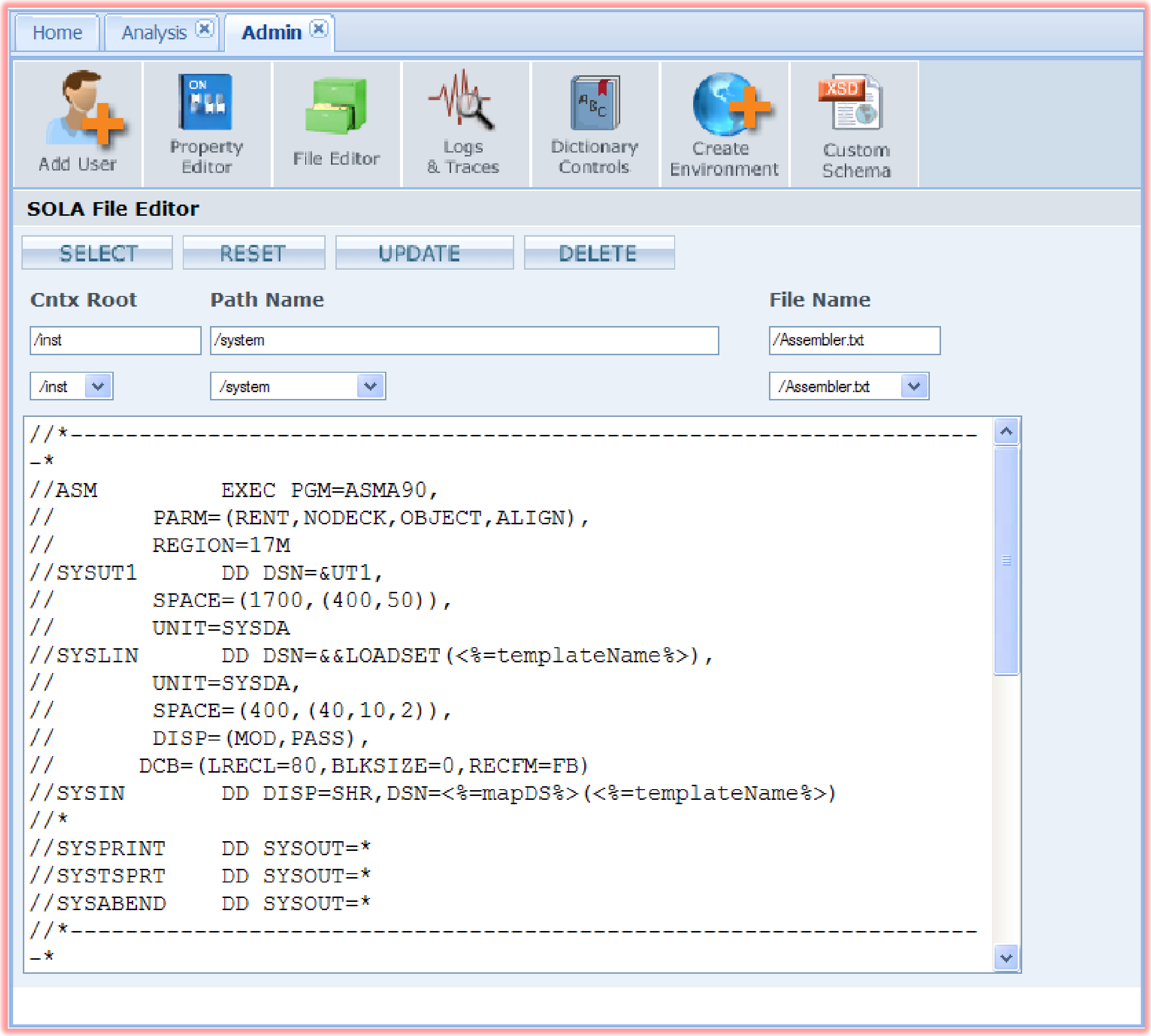


Cntx Root: /inst
Path Name: /system
File Name: /Jobcard.txt

/inst | /system | /Jobcard.txt

```
//<%=userId%>T JOB (1,11111,1111),'JOB CARD',MSGLEVEL=(1,1),
//   CLASS=A,MSGCLASS=H,USER=<%=userId%>,NOTIFY=<%=userId%>
```

To modify the job that assembles the template, you modify Assembler.txt.



Cntx Root: /inst
Path Name: /system
File Name: /Assembler.txt

```
//*-----------------------------------------------------------------
-*
//ASM         EXEC  PGM=ASMA90,
//         PARM=(RENT,NODECK,OBJECT,ALIGN),
//         REGION=17M
//SYSUT1      DD  DSN=&UT1,
//         SPACE=(1700,(400,50)),
//         UNIT=SYSDA
//SYSLIN      DD  DSN=&&LOADSET(<%=templateName%>),
//         UNIT=SYSDA,
//         SPACE=(400,(40,10,2)),
//         DISP=(MOD,PASS),
//        DCB=(LRECL=80,BLKSIZE=0,RECFM=FB)
//SYSIN       DD  DISP=SHR,DSN=<%=mapDS%>(<%=templateName%>)
//*
//SYSPRINT    DD  SYSOUT=*
//SYSTSPRT    DD  SYSOUT=*
//SYSABEND    DD  SYSOUT=*
//*-----------------------------------------------------------------
-*
```

SOLA doesn’t include file promoteJCL.txt. If you wish to integrate SOLA with your change management system then you will need to create a file by that name.

When editing JCL files, you can create your own custom variables using the Custom Schema panel of the admin console. There are also preset variables available for you to use.

The following is a list of preset variables that can be used when customizing JCL:

- Promote and Demote Only
  - fromEnvironment
  - toEnvironment
  - fromEnvironSeq
  - toEnvironSeq
- Project
  - division

  - firstNm
  - lastNm
  - workPhone
  - cellPhone
  - homePhone
  - email
- Program
  - programNm
  - classNm
  - override
  - programType
  - language
  - naturalLib
  - commarea
  - listDs
  - region
  - subType
  - trans
  - term
  - channelName
  - inputContainer
  - outputContainer
  - errorContainer
  - copyDs
  - loadDs
  - templatePs
  - importedOn
- Method
  - methodNm
  - templateNm
  - templatePDS
  - loadDs
- Environment
  - environment
  - environSeq
- User
  - user

- division - optional field
- firstNm - required field
- lastNm - required field
- workPhone - required field
- registered - not updatable
- email - optional
- templatePDS
- loadDs

The JCL constructor sweeps through the properties in the following hierarchy:

Method -> Program -> Project -> Environment -> User

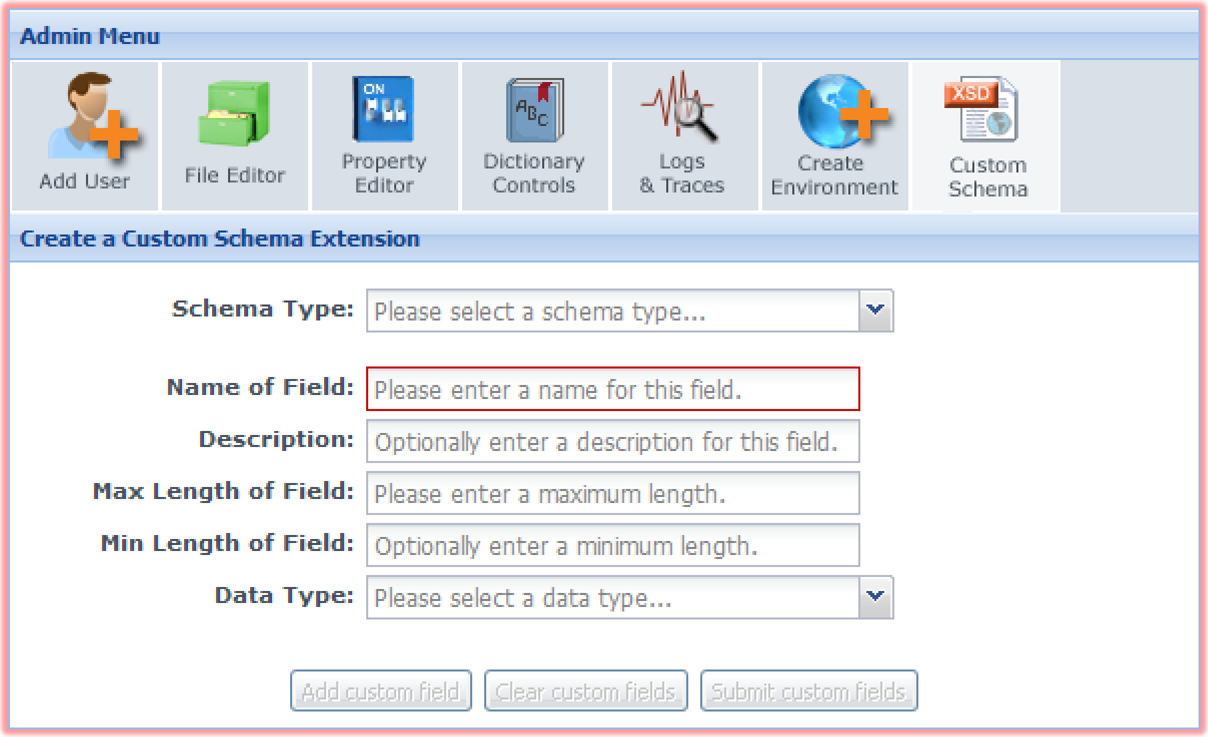
Therefore, if a property exists for both method and program, the JCL constructor will find that property in method first and apply it there (only). The property will not make it to the program.

Installations that need custom properties like 'PackageID' to be used as a part of the promoteJCL.txt

Custom Schemas

The SOLA 6.0 database is an extensible XML database. You can specify the attributes that are stored in that database by modifying **Custom Schemas** for several of the tables in the database.

The custom schema panel allows you to create custom properties for projects, programs, methods, users, environments and more.



Schema Type: Please select a schema type...

Name of Field: Please enter a name for this field.

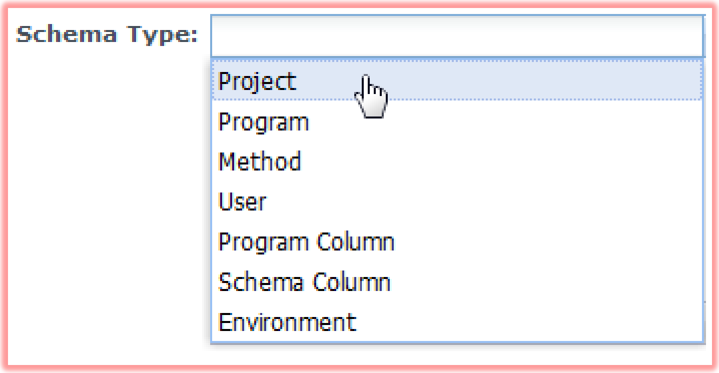
Description: Optionally enter a description for this field.

Max Length of Field: Please enter a maximum length.

Min Length of Field: Optionally enter a minimum length.

Data Type: Please select a data type...

To begin using this panel, first select which type of object you would like to add custom properties to by using the **Schema Type** menu.



Options are:

- **Project:** pick this option to create custom properties for projects. These properties will apply to all projects and will be appended to the default properties.

- **Program:** pick this option to create custom properties for programs. These properties will apply to all programs in every project and will be appended to the default properties.

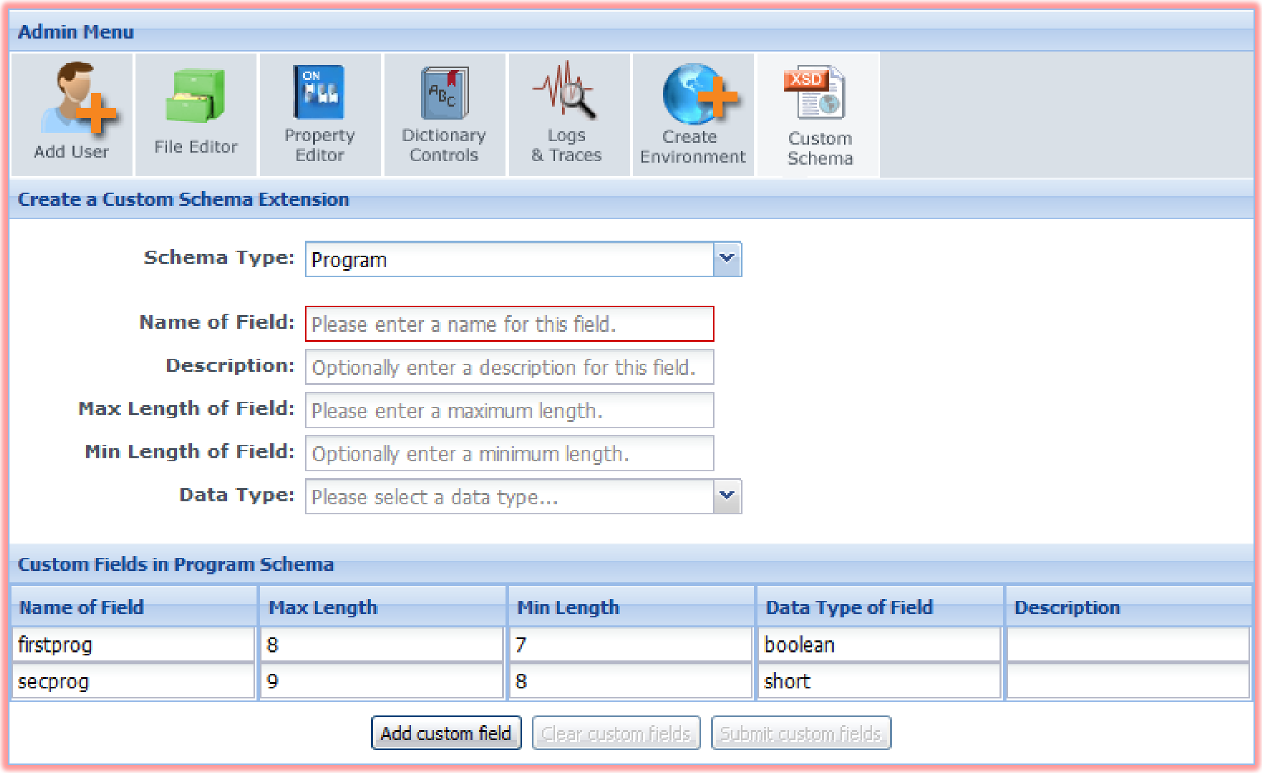
- **Method:** pick this option to create custom properties for methods. These properties will apply to all methods and will be appended to the default properties.

- **User:** pick this option to create custom properties for user accounts. These properties will apply to all users and will be appended to the default properties.

- **Program Column:** pick this option to create custom properties for program columns. These properties will apply to all programs in every project and will be appended to the default properties.

- **Schema Column:**

- **Environment:** pick this option to create custom properties for SOLA Developer environments (e.g. T, S, P, etc.). These properties will apply to all SOLA Developer environments and will be appended to the default properties.



Create a Custom Schema Extension

Schema Type: Program

Name of Field: Please enter a name for this field.

Description: Optionally enter a description for this field.

Max Length of Field: Please enter a maximum length.

Min Length of Field: Optionally enter a minimum length.

Data Type: Please select a data type...

Custom Fields in Program Schema

Name of Field	Max Length	Min Length	Data Type of Field	Description
firstprog	8	7	boolean	
secprog	9	8	short	

The properties are organized under columns. These columns correspond to the value fields in the top part of the panel.

- **Name of Field:** the name of the custom property.

- **Max Length:** the property's maximum length in bytes.
- **Min Length:** the property's minimum length in bytes.
- **Data Type of Field:** the type of data the property can contain. Options are string, int, short and Boolean.
- **Description:** a free form, optional description.

You can add one or more properties by entering values in the value fields. When you have entered all required information, click **Add Custom Field**. Your new property will be displayed under the property columns. If you wish to erase the properties you've added, you can click **Clear Custom Fields**.

Once you are finished adding properties, click **Submit Custom Fields** to finalize your changes. If you close the panel before clicking the submit button, your changes will be lost.

Production Moves

SOLA maintains a version control system for all of the elements in the SOLA Directory. The version control system allows for the existence of multiple versions of a program – for example the version of the program in test can be different from the version in production. The version control system is based on a multiple environment paradigm.

Environments are user defined, as are the promotion/demotion paths between them. For example, you could create three environments, as shown below.

- **Test**
- **Stage**
- **Prod**

Initially when a web service is created, it is created in the Test environment. Test is the only environment where you can make changes to a web service. SOLA provides the ability to "Promote" a program from Test to Stage to Prod and to "Demote" a program from Stage or Prod.

With the new "Temporal Database" design that is integral to SOLA 6.0, the prior method of service versioning is obsolete.

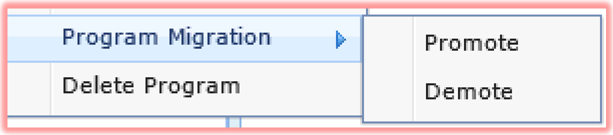
Promote Service

To promote a web service, select the environment you wish to promote from in the drop-down list, then right click on the program name in the program list and select **Promote Program** from the pop-up menu.

The Promote option will cause SOLA to change the status of the Directory entries for your program from Test to Stage.

Promoting the program once more will advance it from Stage to Production.

If you find a bug that needs to be fixed, you can demote the program back to the Test environment. To do so, click on the program name and select **Demote Program** from the pop-up menu.

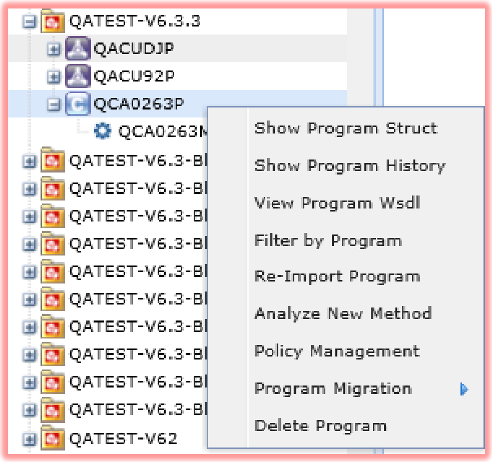


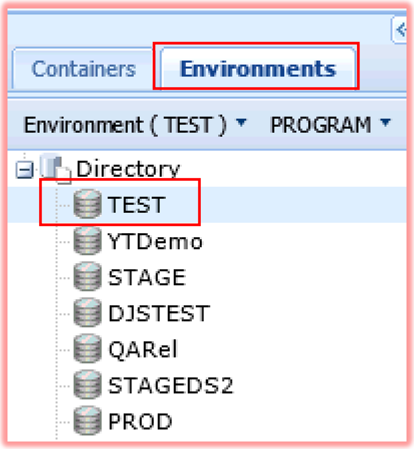
Submitting JCL for Promoted Web Services

If your promotion process requires the submission of JCL, you can create a promoteJcl.txt file that will automatically be submitted when you promote a method to either stage or production.

Promotion/Retrieval Rules

When you promote a program from one environment to another then the setting on attribute 'actionOnPromote' of the source environment is used to determine if the program needs to be '**M**oved' or '**C**opied'.

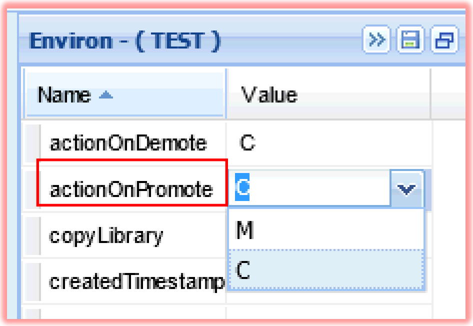




Containers | **Environments**

Environment (TEST) ▾ PROGRAM ▾

- Directory
  - TEST
  - YTDemo
  - STAGE
  - DJSTEST
  - QARel
  - STAGEDS2
  - PROD

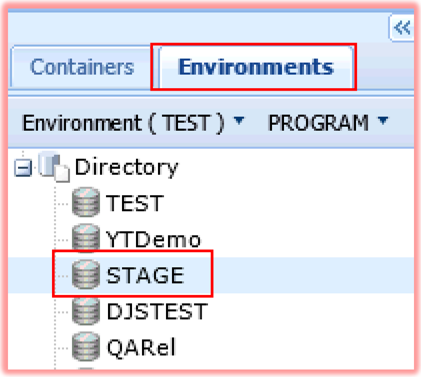


Environ - (TEST)

Name ▲	Value
actionOnDemote	C
actionOnPromote	C ▾
copyLibrary	M
createdTimestamp	C

If it is "**M**oved" then the program's environment is changed, and the program no longer exists under the old environment. If it is copied then the copy of the program is retained in the source environment.

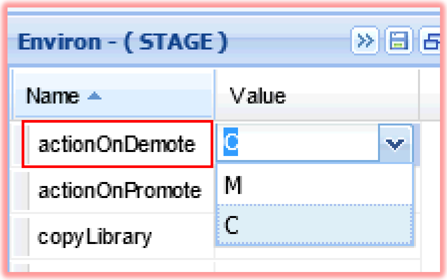
When you demote a program from one environment to another then the setting on attribute 'actionOnDemote' of the target environment is used to determine if the program needs to be '**M**oved' or '**C**opied'. Only exception to this rule is demoting from highest environment (Prod). The Demote from highest environment is always a **C**opy.



Containers | **Environments**

Environment (TEST) ▾ PROGRAM ▾

- Directory
  - TEST
  - YTDemo
  - STAGE
  - DJSTEST
  - QARel



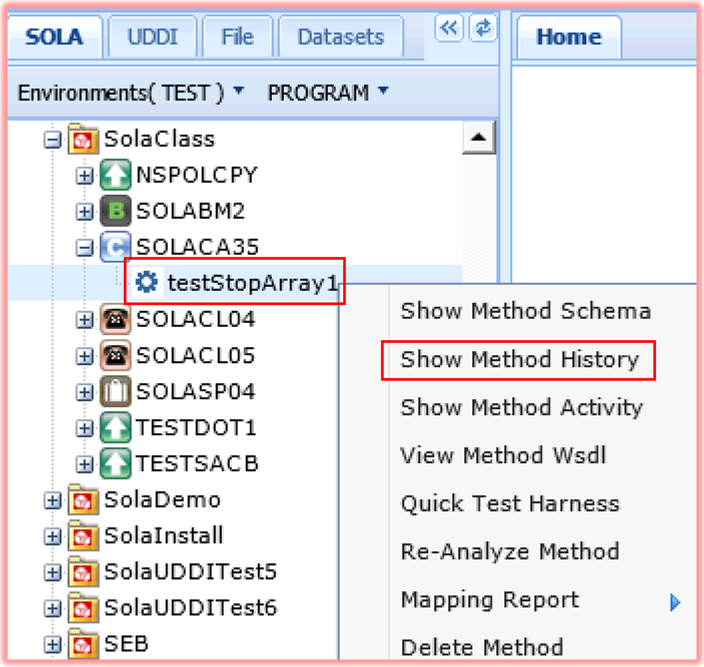
Environ - (STAGE)

Name ▲	Value
actionOnDemote	C ▾
actionOnPromote	M
copyLibrary	C

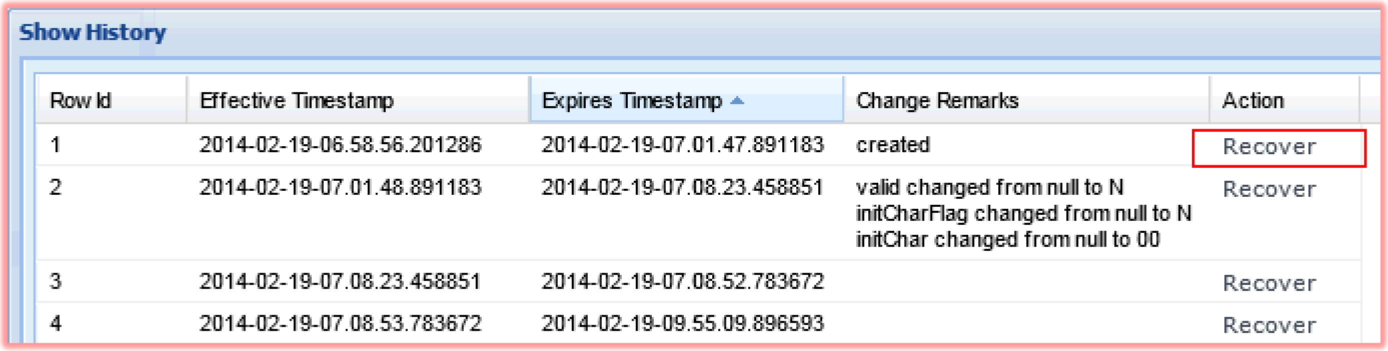
These features enable having different versions of a program in Production and development.

Backup and Restore Methods

Each time a method is modified, the prior version of the method is backed up in the SOLA Directory. You can view the backup summary by right-clicking on a method and choosing **Show Method History** from the pop-up menu.



Doing so will display all of the copies of the method in the directory.



Show History

Row Id	Effective Timestamp	Expires Timestamp	Change Remarks	Action
1	2014-02-19-06.58.56.201286	2014-02-19-07.01.47.891183	created	Recover
2	2014-02-19-07.01.48.891183	2014-02-19-07.08.23.458851	valid changed from null to N initCharFlag changed from null to N initChar changed from null to 00	Recover
3	2014-02-19-07.08.23.458851	2014-02-19-07.08.52.783672		Recover
4	2014-02-19-07.08.53.783672	2014-02-19-09.55.09.896593		Recover

Select ‘Recover’ option to restore older version of methods. If there is any version of method that only have some attribute changes to Method without any changes to the input/output schema layout then ‘Recover’ option will indicate that the version is not recoverable.

SOLA Directory and File System

Backups

The SOLA Directory is a DB2 Database. The majority of the tables in the database are used to keep the metadata associated with creating services. A number of the tables have operational uses for managing certificates, policies and security. Some other tables are used for managing SOLA operations (e.g. maintaining lists of SOLA managed environments), yet others are used for storing metrics and logs such as error or trace.

The database will need to be maintained frequently using standard utilities such as Image Copy and Online Reorg. We recommend that you add maintenance jobs for the SOLA Directory to your job scheduler and run those jobs frequently.

SOLA Developer, SOLA's "front end", runs on a J2EE server and uses a file system. This file system's root directory can be defined as a custom property (on the J2EE sever) called "SOLARoot". If this custom property is not defined, SOLA will use a default directory structure.

Any files maintained on the J2EE server's file-system are not critical and SOLA can recover automatically if this file system is lost. Nevertheless, it is recommended that you back up the file system periodically.

Database

The SOLA 6.0 DB2 database is entirely different from the SOLA 5.x DB2 database. It comprises 28 tables (including the sample application table TBXMLWGT). SOLA 6.0 maintains historical data by using a temporal database design.

The tables in the SOLA 6.0 database are:

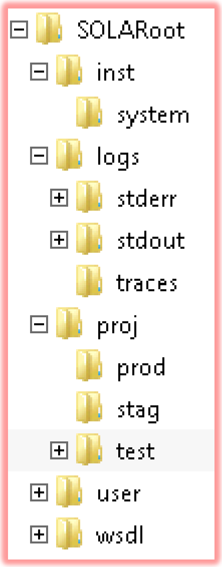
Name	Description	Maintenance
TBXMLACC	Access Control table	Image Copy regularly
TBXMLASN	Association table – used in combination with TBXMLGRP	Image Copy regularly
TBXMLCER	Certificate table	Image Copy regularly
TBXMLCOM	Commarea table	Image Copy regularly
TBXMLENV	Environment table	Image Copy regularly
TBXMLEXT	User Exits	Image Copy regularly
TBXMLGRP	Group table	Image Copy regularly
TBXMLIPA	IP Address table	Image Copy regularly
TBXMLLOG	Log table	Image Copy regularly
TBXMLMAP	BMS Map table	Image Copy regularly

TBXMLMON	Monitor table	Image Copy regularly
TBXMLMSK	Mask table	Image Copy regularly
TBXMLMTD	Method table	Image Copy regularly
TBXMLMTL	Method column table	Image Copy regularly
TBXMLMTS	3270 fields table	Image Copy regularly
TBXMLOFT	Overflow table	Image Copy regularly
TBXMLPGM	Program table	Image Copy regularly
TBXMLPOL	Policy table	Image Copy regularly
TBXMLPRJ	Project table	Image Copy regularly
TBXMLSCH	Schema table	Image Copy regularly
TBXMLSPT	Field split table	Image Copy regularly
TBXMLTOR	SOLA Container table	Image Copy regularly
TBXMLUAC	User Activity child table	Image Copy regularly
TBXMLUAP	User Activity parent table	Image Copy regularly
TBXMLUAR	User Activity request table	Image Copy regularly
TBXMLUSR	Authorized User table	Image Copy regularly
TBXMLWGT	Sample application data table	Image Copy occasionally

File System

The custom property “SOLARoot”, defined to the J2EE Server, specifies the path to or location of the SOLA file system. These files are not critical and SOLA can recover automatically even when this file system is lost, with the exception of customized parameters.

The file system has the following structure:



The following system files are stored under SOLARoot\inst\system:

- **indexpage.html:** The html for the SOLA home page. Customizing the SOLA home page is discussed in section **Customization of the SOLA Home Page** on page 9.
- **Endpoints.xml:** Updating the endpoints.xml file is discussed below.
- **Debugging.xml:** Updating debugging.xml is discussed in section
- **Property** Changes on page 40.

Since this folder contains all customizations of the SOLA IDE, it is recommended that you back it up periodically.

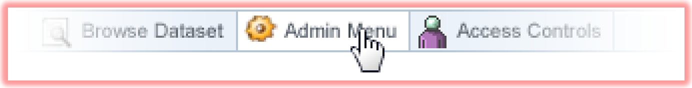
SOLA Developer Logs

Installation file SOLARoot\inst\system\Debugging.xml has a property 'debug' whose setting will define the extent of logging that happens on the system. The valid values of 'debug' property are :

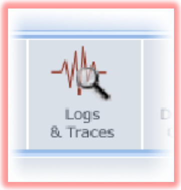
I – Informational and higher severity messages to be logged
W- Warning and higher severity messages and higher to be logged
E – Only messages to be logged

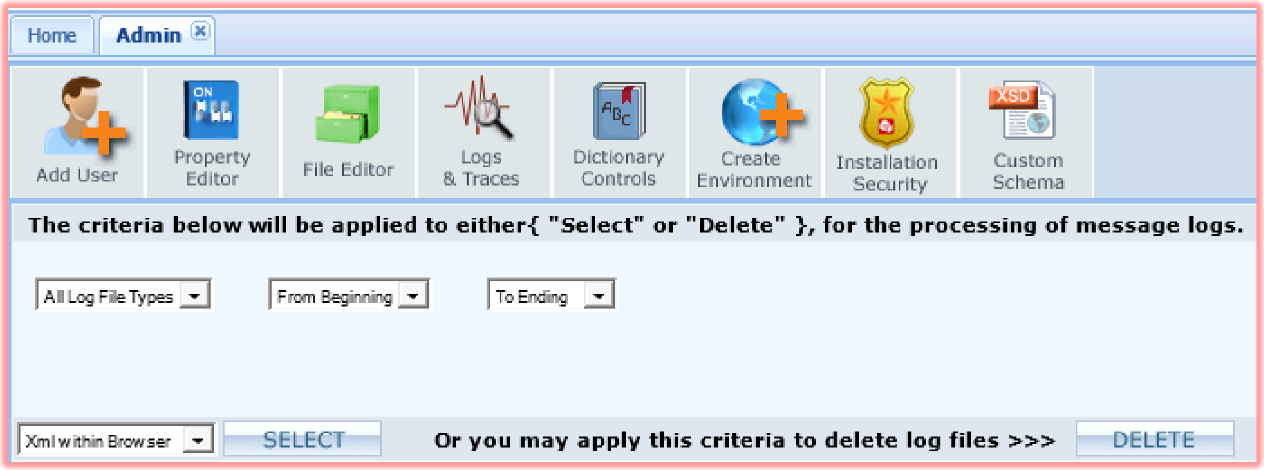
Folders 'stdout' and 'stderr' under \SOLARoot\logs directory holds all the Informational and error messages. Within these fodlers, log data is organized in date wise folders for easy maintenance and lookup.

To view or clean the SOLA Developer log files, access the SOLA Admin Menu screen by selecting **Admin Menu** from the SOLA Developer menu bar.



Click on Logs and Traces to bring up the Logs and Traces screen

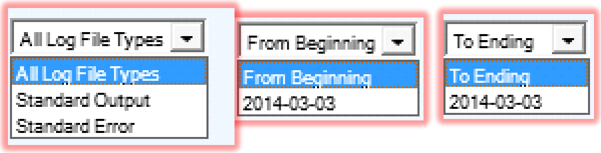




The criteria below will be applied to either{ "Select" or "Delete" }, for the processing of message logs.

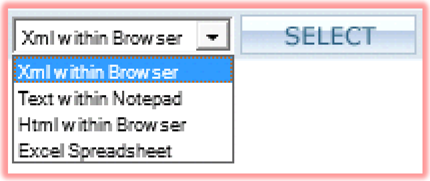
Or you may apply this criteria to delete log files >>>

Using path menus as illustrated below set the filters and press **SELECT** to view the log files.

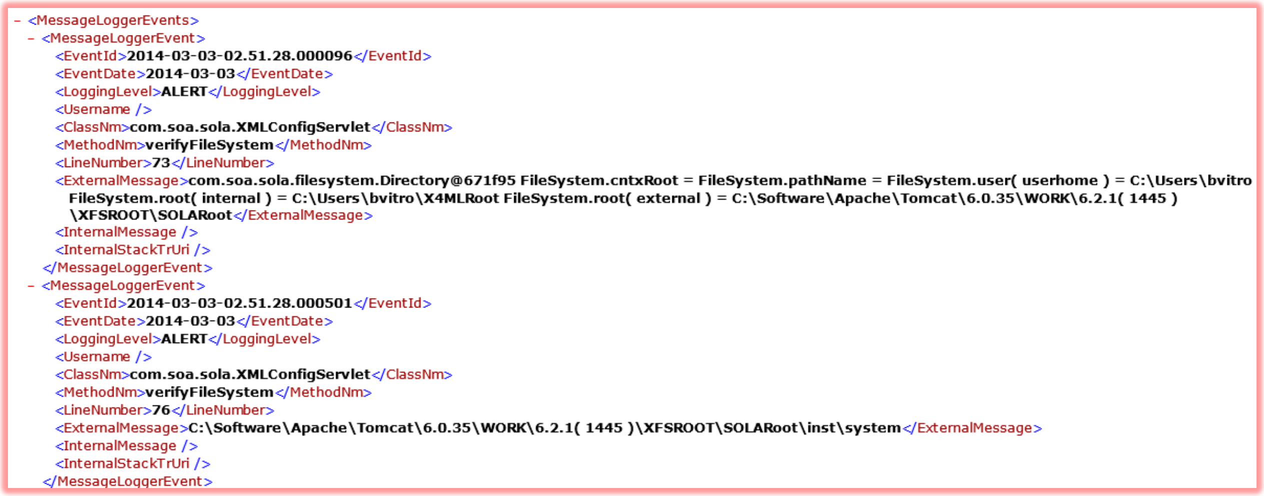


Housekeeping of log files works with the same filters. Setup the filters and press **DELETE** for cleaning up of corresponding matching logs

Rendering of Log data is supported in following 4 formats



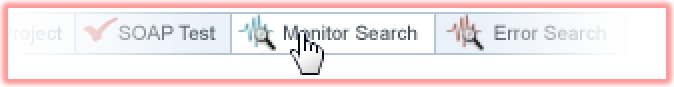
The following is the illustration of Logs rendered in XML format within browser.



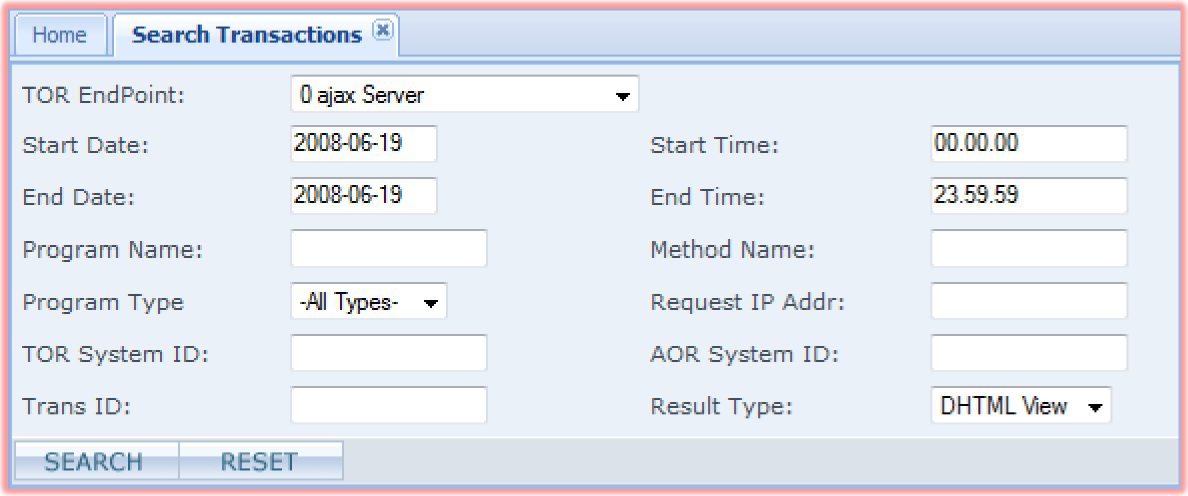
```
- <MessageLoggerEvents>
  - <MessageLoggerEvent>
      <EventId>2014-03-03-02.51.28.000096</EventId>
      <EventDate>2014-03-03</EventDate>
      <LoggingLevel>ALERT</LoggingLevel>
      <Username />
      <ClassNm>com.soa.sola.XMLConfigServlet</ClassNm>
      <MethodNm>verifyFileSystem</MethodNm>
      <LineNumber>73</LineNumber>
      <ExternalMessage>com.soa.sola.filesystem.Directory@671f95 FileSystem.cntxRoot = FileSystem.pathName = FileSystem.user( userhome ) = C:\Users\bvitro
        FileSystem.root( internal ) = C:\Users\bvitro\X4MLRoot FileSystem.root( external ) = C:\Software\Apache\Tomcat\6.0.35\WORK\6.2.1( 1445 )
        \XFSROOT\SOLARoot</ExternalMessage>
      <InternalMessage />
      <InternalStackTrUri />
    </MessageLoggerEvent>
  - <MessageLoggerEvent>
      <EventId>2014-03-03-02.51.28.000501</EventId>
      <EventDate>2014-03-03</EventDate>
      <LoggingLevel>ALERT</LoggingLevel>
      <Username />
      <ClassNm>com.soa.sola.XMLConfigServlet</ClassNm>
      <MethodNm>verifyFileSystem</MethodNm>
      <LineNumber>76</LineNumber>
      <ExternalMessage>C:\Software\Apache\Tomcat\6.0.35\WORK\6.2.1( 1445 )\XFSROOT\SOLARoot\inst\system</ExternalMessage>
      <InternalMessage />
      <InternalStackTrUri />
    </MessageLoggerEvent>
```

Transaction Logs

You can search the transaction logs by clicking the **Monitor Search** button on the button bar.



This will display the monitor search panel.



TOR EndPoint:	0 ajax Server		
Start Date:	2008-06-19	Start Time:	00.00.00
End Date:	2008-06-19	End Time:	23.59.59
Program Name:		Method Name:	
Program Type	-All Types-	Request IP Addr:	
TOR System ID:		AOR System ID:	
Trans ID:		Result Type:	DHTML View

To conduct a search of the transaction log, enter search parameters using the search fields to narrow the scope of your search. You can also conduct a search with the default (mostly blank) settings, though this may take some time to complete and may result in a very long list of transactions.

The following is a description of the search fields:

- **TOR EndPoint:** narrows the search to transactions within a matching TOR region.
- **Start Date and End Date:** the start and end dates are automatically populated with the current date, though these values can be changed if necessary. All transactions are stamped with the date and time at which they take place, and only transactions that took place on or after the start date and on or before the end date will be returned.

- **Program Name:** narrows the search to transactions executing this program.

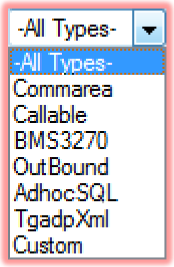
- **TOR System ID:** narrows the search to transactions with a matching TOR system Id.

- **Trans ID:** narrows the search to transactions with a matching transaction Id.

- **Start Time and End Time:** the start and end times are automatically populated with the current system time and can be changed by manually entering a time (hh.mm.ss). All transactions are stamped with the date and time at which they take place, and only transactions that took place at or after the start time and at or before the end time will be returned.

- **Method Name:** narrows the search to transactions generated by the execution of the specified method.

- **Program Type:** narrows the search to transactions initiated by a method executed by the specified program type. Options are All Types, Commarea, Callable, BMS3270, Outbound, AdhocSQL, IMS, TgadpXml or Custom.



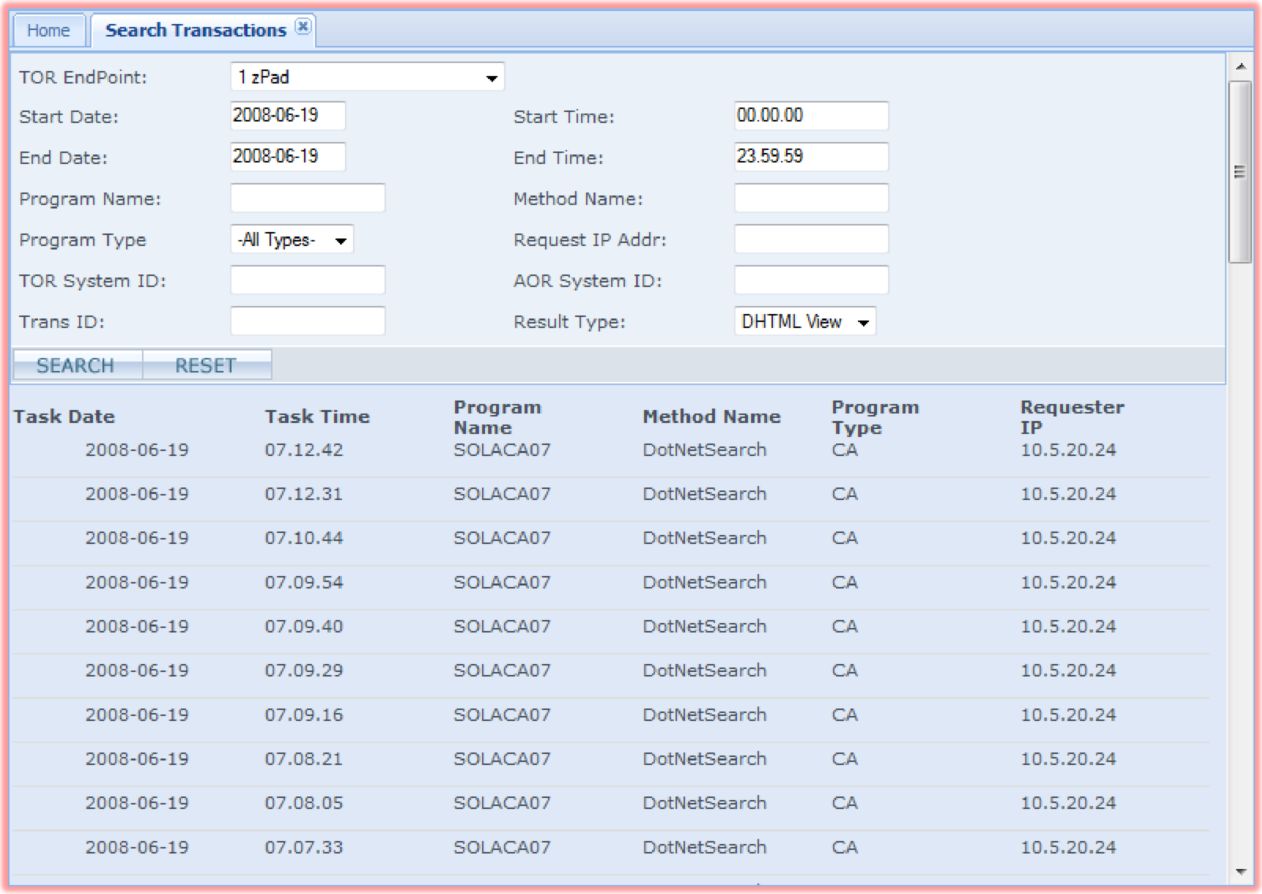
- **Request IP Addr:** narrows the search to transactions generated in response to a request that originated from an IP address matches the specified IP address (if the request came via HTTP).

- **AOR System ID:** narrows the search to transactions with a matching AOR system Id.

- **Result Type:** specifies how the results will be displayed, either as DHTML (normal view) or as an Excel spreadsheet. Selecting Excel will download the results and open MS Excel (if installed), displaying the data in an Excel spreadsheet.

Once you have specified your search parameters, click SEARCH.

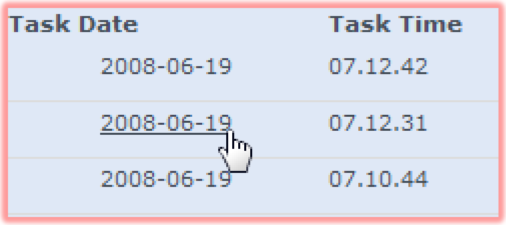
The results of the search will be displayed below the monitor search panel. If the list exceeds the available screen size, then you will need to scroll to see all of the search results.



TOR EndPoint:	1 zPad		
Start Date:	2008-06-19	Start Time:	00.00.00
End Date:	2008-06-19	End Time:	23.59.59
Program Name:		Method Name:	
Program Type	-All Types-	Request IP Addr:	
TOR System ID:		AOR System ID:	
Trans ID:		Result Type:	DHTML View

Task Date	Task Time	Program Name	Method Name	Program Type	Requester IP
2008-06-19	07.12.42	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.12.31	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.10.44	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.09.54	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.09.40	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.09.29	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.09.16	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.08.21	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.08.05	SOLACA07	DotNetSearch	CA	10.5.20.24
2008-06-19	07.07.33	SOLACA07	DotNetSearch	CA	10.5.20.24

The information is organized under a series of columns:



Task Date	Task Time
2008-06-19	07.12.42
2008-06-19	07.12.31
2008-06-19	07.10.44

- **Task Date:** the day the transaction was generated, represented as yyyy-mm-dd. Clicking on the date for a specific transaction displays the search details panel that contains very detailed information about the transaction.

- **Task Time:** the time the transaction was generated, represented as hh.mm.ss.

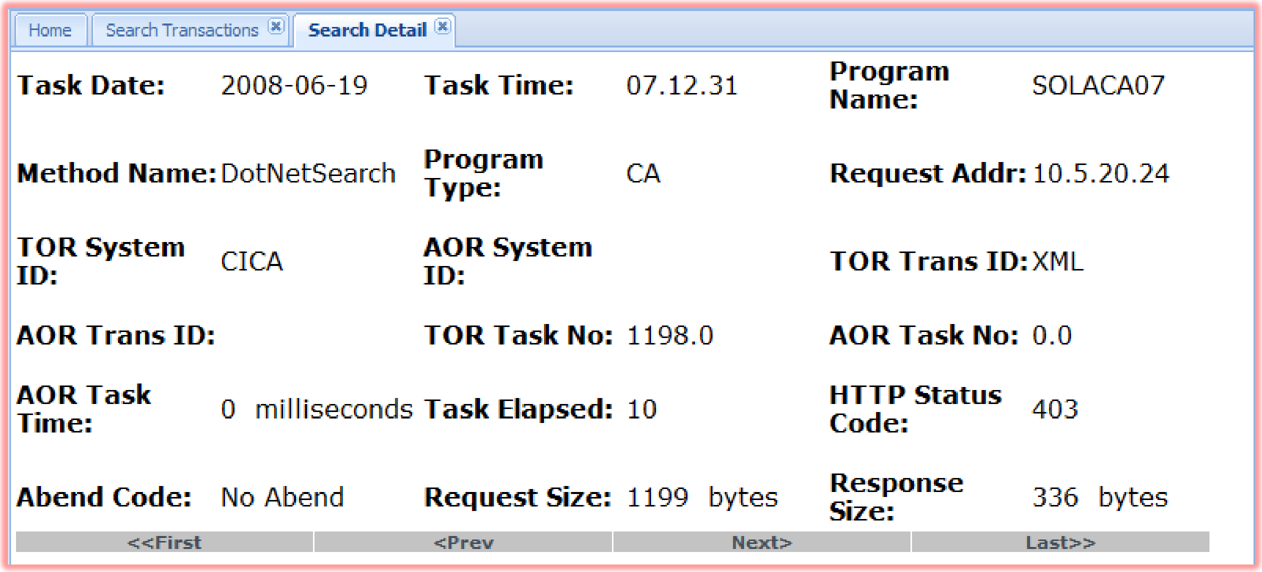
- **Program Name:** the program whose execution generated the transaction.

- **Method Name:** the name of the method whose execution generated the transaction.

- **Program Type:** the category (type) of program whose execution generated the transaction.

- **Requester IP:** the IP Address of the originating request (responsible for executing the method that generated the transaction, if it comes via HTTP).

To get detailed information about a specific transaction, click on the transaction date. This will display the search detail panel.



Task Date:	2008-06-19	**Task Time:**	07.12.31	**Program Name:**	SOLACA07
Method Name:	DotNetSearch	**Program Type:**	CA	**Request Addr:**	10.5.20.24
TOR System ID:	CICA	**AOR System ID:**		**TOR Trans ID:**	XML
AOR Trans ID:		**TOR Task No:**	1198.0	**AOR Task No:**	0.0
AOR Task Time:	0 milliseconds	**Task Elapsed:**	10	**HTTP Status Code:**	403
Abend Code:	No Abend	**Request Size:**	1199 bytes	**Response Size:**	336 bytes

This panel contains detailed information about a specific transaction organized under the following headings:

- **Task Date:** the date (yyyy-mm-dd) of the transaction.

- **Task Time:** the time (hh.mm.ss) of the transaction.

- **Program Name:** the program whose execution generated the transaction.

- **Method Name:** the method whose execution generated the transaction.

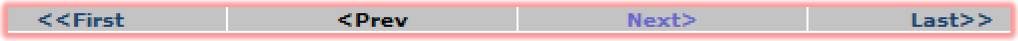
- **Program Type:** the type of the program whose execution generated the transaction.

- **Request Addr:** the IP Address of the originating request (responsible for executing the method that generated the transaction).

- **TOR System ID:** unique identifier for the TOR region where the transaction originated.

- **AOR System ID:** unique identifier for the AOR region where the transaction originated.
- **TOR Trans ID:** unique identifier given to each program that runs in a TOR.
- **AOR Trans ID:** unique identifier given to each program that runs in a AOR.
- **TOR Task No:** unique identifier that is given to each unique instance of a program running in a TOR.
- **AOR Task No:** unique identifier that is given to each unique instance of a program running in a AOR.
- **AOR Task Time:** how long it took to execute the program in the AOR, accurate to +/- 5 milliseconds.
- **Task Elapsed:** the total end to end time (AOR+TOR) that it took to execute the program, accurate to +/- 5 milliseconds.
- **HTTP Status Code:** the HTTP response code generated as a result of the transaction (e.g. 200 – OK, 403 – Auth Failure, etc.)
- **Abend Code:** the mainframe abend code if the program abnormally terminates (i.e. abnormally ends - abends).
- **Request Size:** the size of the input SOAP XML in bytes.
- **Response Size:** the size of the output SOAP XML in bytes.

The links at the bottom of the panel allow you to navigate through all the transactions in the list.



<<First: show details for the first transaction in the list.

<Prev: show details for the previous transaction.

Next>: show details for the next transaction.

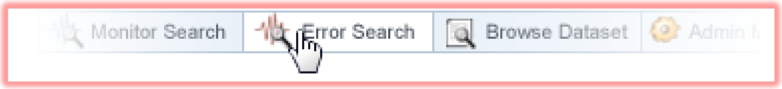
Last>>: show details for the last transaction.

Maintenance

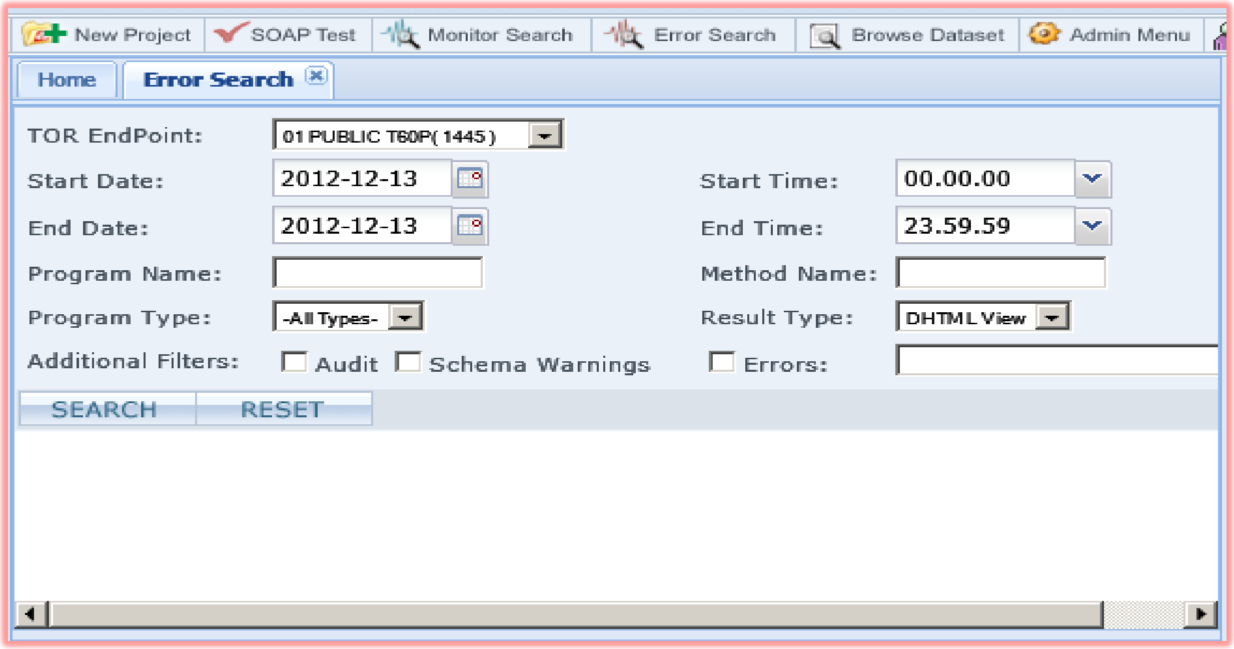
The Transaction Log is a DB2 table, <*qualifier*>.TBXMLMON. We recommend that you maintain the table regularly by purging old data. One way of doing this is to use the DB2 Online Reorg with the "Delete When" option to delete rows that are older than a specified age.

Error Logs

You can search the error logs by clicking the **Error Search** button on the button bar.



This will display the error search panel.



TOR EndPoint: 01 PUBLIC T60P(1445)

Start Date: 2012-12-13 Start Time: 00.00.00

End Date: 2012-12-13 End Time: 23.59.59

Program Name: Method Name:

Program Type: -All Types- Result Type: DHTML View

Additional Filters: □ Audit □ Schema Warnings □ Errors:

SEARCH RESET

To conduct a search of the error log, enter search parameters using the search fields to narrow the scope of your search. You can also conduct a search with the default (mostly blank) settings.

The following is a description of the search fields:

- **TOR EndPoint:** narrows the search to errors generated within a matching TOR region.

- **Start Date and End Date:** the start and end dates are automatically populated with the current date, though these values can be changed if necessary. All errors are stamped with the date and time at which they take place, and only errors that took place on or after the start date and on or before the end date will be returned.

- **Start Time and End Time:** the start and end times are automatically populated with the current system time and can be changed by manually entering a time (hh.mm.ss). All errors are stamped with the date and time at which they take place, and only errors that took place at or after the start time and at or before the end time will be returned.

- **Program Name:** narrows the search to errors generated by the specified program.

- **Method Name:** narrows the search to errors generated by the specified method.

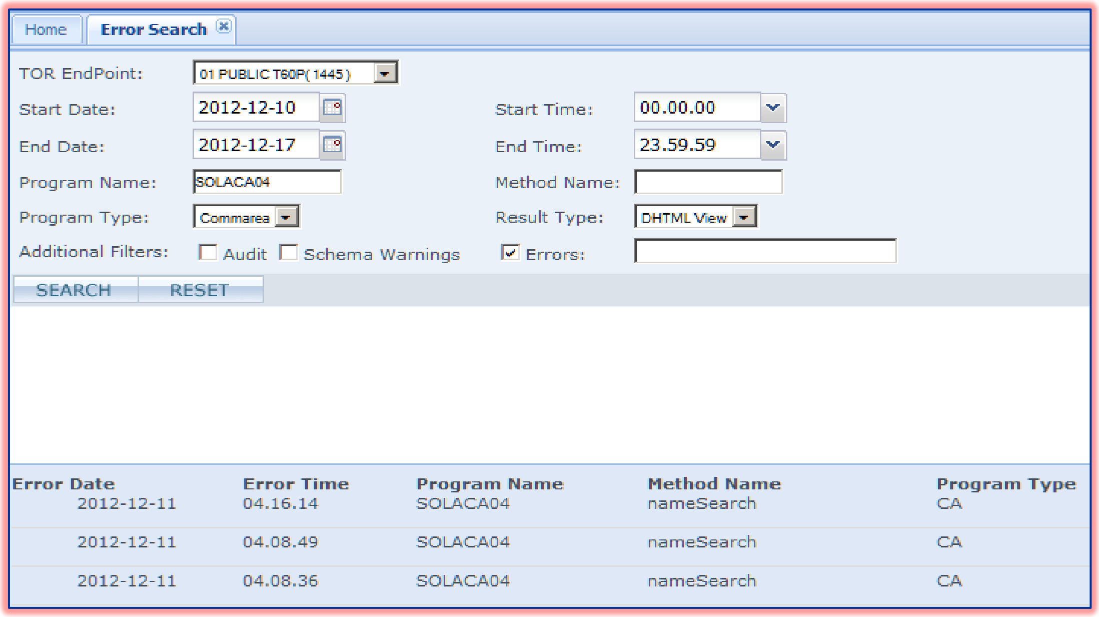
- **Program Type:** narrows the search to errors generated by a method executed by the specified program type. Options are All Types, Commarea, Callable, BMS3270, Outbound, AdhocSQL, TgadpXml or Custom.

- **Result Type:** specifies how the results will be displayed, either as html (normal view) or as an Excel spreadsheet. Selecting Excel will download the results and open MS Excel (if installed), displayed the data in an Excel spreadsheet.

- **Additional Filters:** narrows the search to include only Audit Information, Schema Warnings or specific Error codes.

Once you have specified your search parameters, click SEARCH.

The results of the search will be displayed below the error search panel. If the list exceeds the available screen size, then you will need to scroll to see all of the search results.

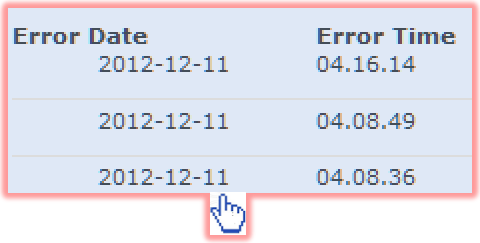


TOR EndPoint:	01 PUBLIC T60P(1445)		
Start Date:	2012-12-10	Start Time:	00.00.00
End Date:	2012-12-17	End Time:	23.59.59
Program Name:	SOLACA04	Method Name:	
Program Type:	Commarea	Result Type:	DHTML View
Additional Filters:	□ Audit □ Schema Warnings	☑ Errors:	

Error Date	Error Time	Program Name	Method Name	Program Type
2012-12-11	04.16.14	SOLACA04	nameSearch	CA
2012-12-11	04.08.49	SOLACA04	nameSearch	CA
2012-12-11	04.08.36	SOLACA04	nameSearch	CA

The information is organized under a series of columns:

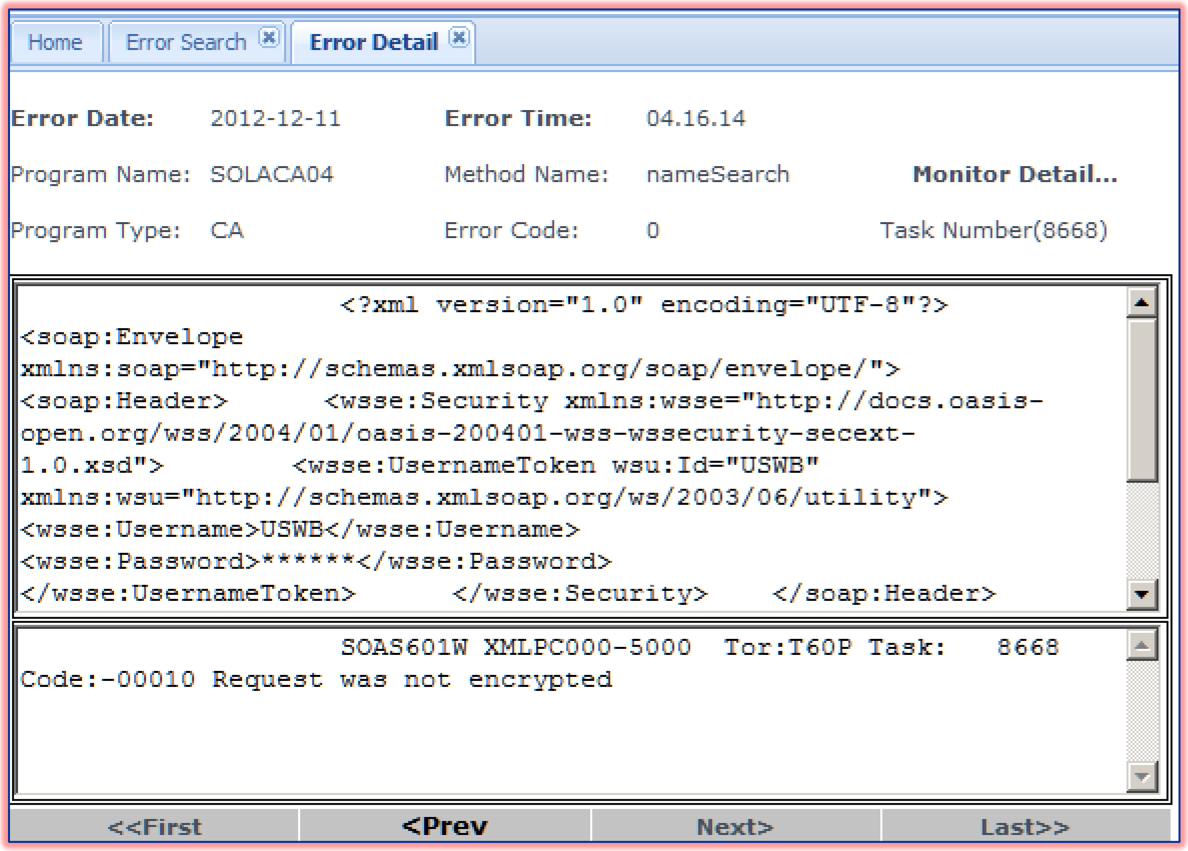
- **Error Date:** the day the error was generated, represented as yyyy-mm-dd. Clicking on the date for a specific error displays the search details panel that contains very detailed information about the error.



Error Date	Error Time
2012-12-11	04.16.14
2012-12-11	04.08.49
2012-12-11	04.08.36

- **Error/Task Time:** the time the error was generated, represented as hh.mm.ss.
- **Program Name:** the program that generated the error.
- **Method Name:** the name of the method that generated the error.
- **Program Type:** the category (type) of program that generated the error.

To get detailed information about a specific error, click on the error date. This will display the search detail panel.



Error Date:	2012-12-11	**Error Time:**	04.16.14	
Program Name:	SOLACA04	Method Name:	nameSearch	**Monitor Detail...**
Program Type:	CA	Error Code:	0	Task Number(8668)

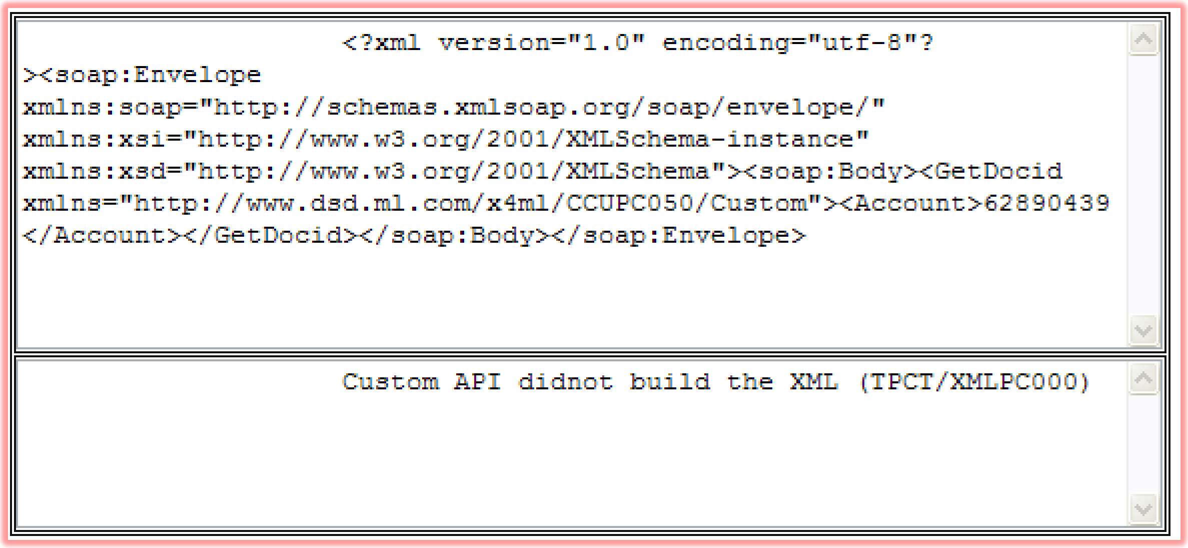
```
                    <?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Header>      <wsse:Security xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">        <wsse:UsernameToken wsu:Id="USWB"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
<wsse:Username>USWB</wsse:Username>
<wsse:Password>******</wsse:Password>
</wsse:UsernameToken>      </wsse:Security>    </soap:Header>
```

```
                    SOAS601W XMLPC000-5000  Tor:T60P Task:   8668
Code:-00010 Request was not encrypted
```

This panel contains detailed information about a specific error organized under the following headings:

- **Error Date:** the date (yyyy-mm-dd) of the error.
- **Error Time:** the time (hh.mm.ss) of the error.
- **Program Name:** the parent program of the method that caused the error.
- **Method Name:** the method that caused the error.
- **Monitor Detail:** provides detailed information about a specific transaction. For further detailed description of this information see the Resource Manager Users Guide.
- **Program Type:** the category of the parent program of the method whose execution caused the error.
- **Error Code:** the error code of the generated error.
- **Task Number:** the TOR task number of the task that caused the error. the AOR, accurate to +/- 5 milliseconds.

This panel contains an error display field that contains additional debugging information.



```
                    <?xml version="1.0" encoding="utf-8"?
><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetDocid
xmlns="http://www.dsd.ml.com/x4ml/CCUPC050/Custom"><Account>62890439
</Account></GetDocid></soap:Body></soap:Envelope>
```

```
                    Custom API didnot build the XML (TPCT/XMLPC000)
```

This field is divided into two panes. The bottom pane displays the mainframe error message, while the top pane displays the input XML that caused the error.

The links at the bottom of the panel allow you to navigate through all the errors in the list.

<<First: show details for the first error in the list.

<Prev: show details for the previous error.

Next>: show details for the next error.

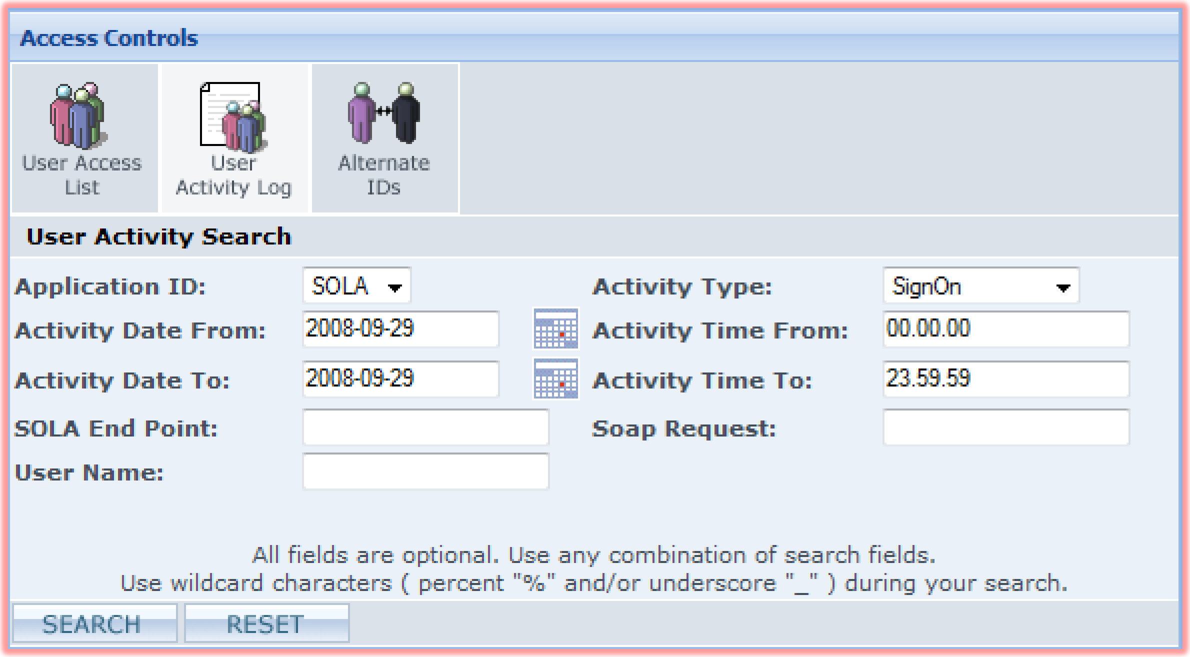
Last>>: show details for the last error.

Maintenance

The Error Log is a DB2 table, <*qualifier*>.TBXMLLOG. We recommend that you maintain the table regularly by purging old data. One way of doing this is to use the DB2 Online Reorg with the “Delete When” option to delete rows that are older than a specified age.

User Activity Log

The user activity log panel is used to search through SOLA's user activity log in the same way that the monitor search panel (page 86) is used to search SOLA's transaction log.



User Activity Search

Application ID:	SOLA	Activity Type:	SignOn
Activity Date From:	2008-09-29	Activity Time From:	00.00.00
Activity Date To:	2008-09-29	Activity Time To:	23.59.59
SOLA End Point:		Soap Request:	
User Name:			

All fields are optional. Use any combination of search fields.
Use wildcard characters (percent "%" and/or underscore "_") during your search.

To conduct a search of the activity log, enter search parameters using the search fields to narrow the scope of your search. You can also conduct a search with the default (mostly blank) settings, though this may take some time to complete and may result in a very long list of activities.

The following is a description of the search fields:

- **Application ID:** currently, the only option is SOLA.

- **Activity Type:** narrows the search to activities of the specified type. Options are:
  - **SignOn:** user sign-on.
  - **Error Search:** error log search.
  - **Monitor Search:** monitor (transaction) search.
  - **Testing:** quick test or raw test.
  - **Import:** importing a program.
  - **Analysis:** method analysis.
  - **Delete:** deletion of a project, program or method.

- **Activity Date From and Activity Date To:** the start (activity date from) and end (activity date to) dates are automatically populated with the current date and can be changed by manually entering a date (yyyy-mm-dd). All activities are stamped with the date and time at which they take place, and only activities that took place on or after the start date and on or before the end date will be returned.

- **Activity Time From and Activity Time To:** the start and end times are automatically populated with the current system time and can be changed by manually entering a time (hh.mm.ss). All activities are stamped with the date and time at which they take place, and only activities that took place at or after the start time and at or before the end time will be returned.

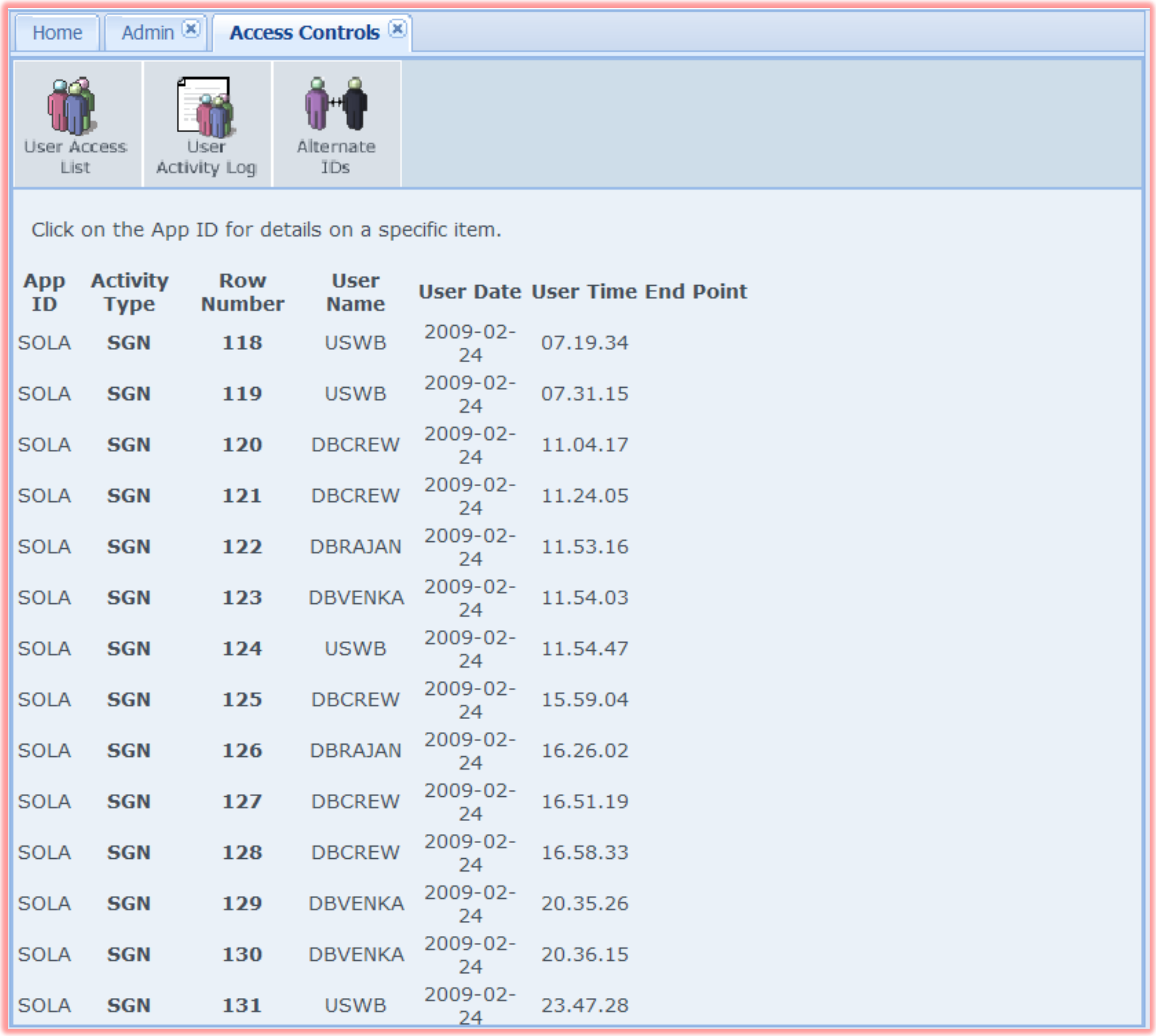
- **SOLA End Point:** narrows the search to activities that involve a request with the specified end point.

- **SOAP Request:** if an activity involves a SOAP request sent through the SOLA website, then this field can be used to narrow the search based on a part of that SOAP request. For example, if you populate this field with the word "SOLA", then any activity that involved a SOAP request with the word SOLA in any context will be returned (provided it matches any other search parameters that are specified).

- **User Name:** narrows the search to the activities of the specified user.

Once you have specified your search parameters, click SEARCH.

The results of the search will be displayed below the activity search panel. If the list exceeds the available screen size, then you will need to scroll to see all of the search results.



Click on the App ID for details on a specific item.

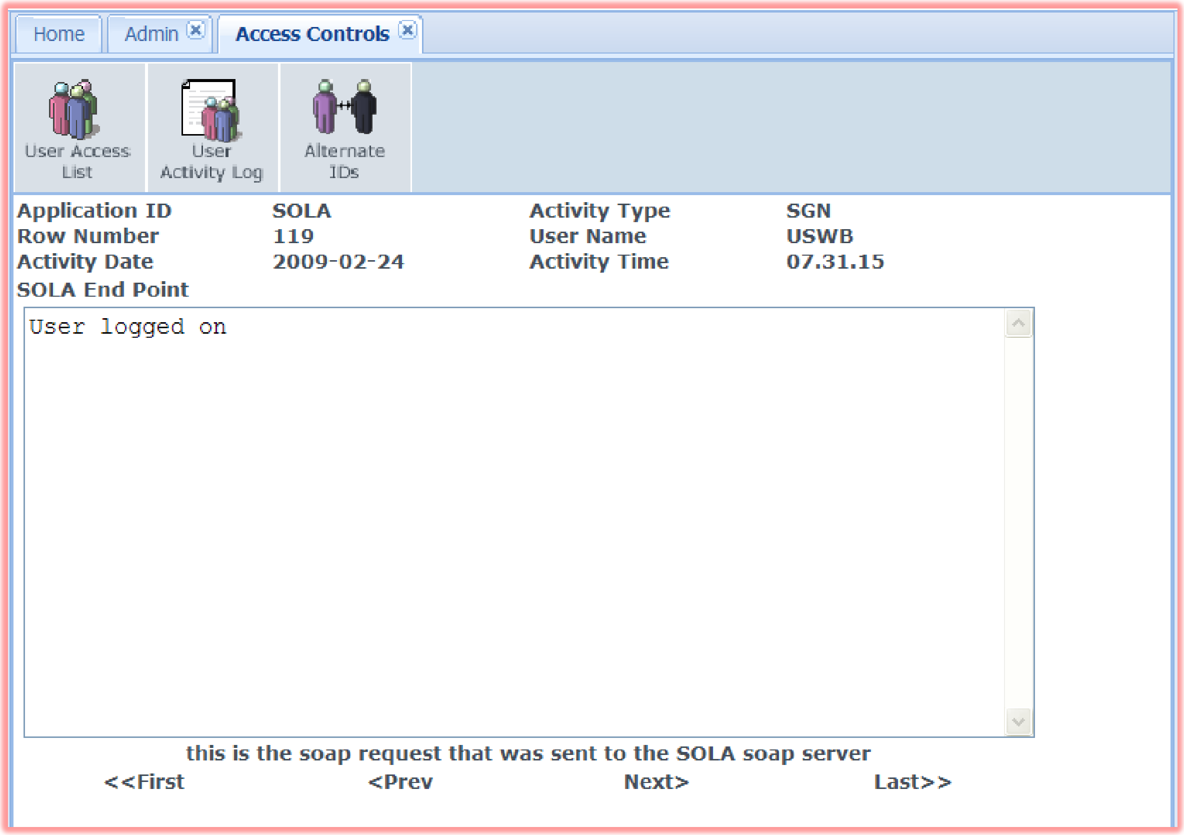
App ID	Activity Type	Row Number	User Name	User Date	User Time	End Point
SOLA	SGN	118	USWB	2009-02-24	07.19.34	
SOLA	SGN	119	USWB	2009-02-24	07.31.15	
SOLA	SGN	120	DBCREW	2009-02-24	11.04.17	
SOLA	SGN	121	DBCREW	2009-02-24	11.24.05	
SOLA	SGN	122	DBRAJAN	2009-02-24	11.53.16	
SOLA	SGN	123	DBVENKA	2009-02-24	11.54.03	
SOLA	SGN	124	USWB	2009-02-24	11.54.47	
SOLA	SGN	125	DBCREW	2009-02-24	15.59.04	
SOLA	SGN	126	DBRAJAN	2009-02-24	16.26.02	
SOLA	SGN	127	DBCREW	2009-02-24	16.51.19	
SOLA	SGN	128	DBCREW	2009-02-24	16.58.33	
SOLA	SGN	129	DBVENKA	2009-02-24	20.35.26	
SOLA	SGN	130	DBVENKA	2009-02-24	20.36.15	
SOLA	SGN	131	USWB	2009-02-24	23.47.28	

The information is organized under a series of columns:

- **App ID:** the application involved in the activity. Clicking on the application ID for a specific activity displays the search details panel that contains very detailed information about the activity.
- **Activity Type:** the type of activity. Option are:
  - **SGN:** user sign-on.
  - **MON:** monitor (transaction) search.
  - **LOG:** error log search.
  - **TST:** quick test or raw test.
  - **DEL:** deletion of a project, program or method.
  - **TRC:** a trace initiated by an administrator.

- **Row Number:** each activity is assigned a sequence number, which is displayed here.
- **User Name:** the user involved in the activity.
- **User Date:** the day the activity took place expressed as yyyy-mm-dd.
- **User Time:** the time the activity took place expressed as hh.mm.ss.
- **End Point:** the end point in which the activity took place.

To get detailed information about a specific activity, click on the activity's application ID. This will display the search details panel.



Application ID	**SOLA**	**Activity Type**	**SGN**
Row Number	**119**	**User Name**	**USWB**
Activity Date	**2009-02-24**	**Activity Time**	**07.31.15**
SOLA End Point			

```
User logged on
```

this is the soap request that was sent to the SOLA soap server

The search details is a series of fields that contain data about a specific user activity, along with a large field that contains the soap request that was sent to the SOLA soap server as a part of the activity (if there was one).

The information is organized under the following headings:

- **Application ID:** currently, the only option is SOLA.
    - **Activity Type:** narrows the search to activities of the specified type. Options are:
    - **SGN:** user sign-on.
    - **MON:** monitor (transaction) search.
    - **LOG:** error log search.
    - **TST:** quick test or raw test.
    - **TRC:** a trace initiated by an administrator.
- **User Name:** the user account that triggered the activity.
- **Activity Date:** the date (yyyy-mm-dd) of the activity.
- **Activity Time:** the time (hh.mm.ss) of the activity.
- **SOLA End Point:** the mainframe end point where the activity took place.

The links at the bottom of the panel allow you to navigate through all the activities in the list.

<<First: show details for the first activity in the list.

<Prev: show details for the previous activity.

Next>: show details for the next activity.

Last>>: show details for the last activity.

Maintenance

No maintenance is required for the User Activity log. It is implemented as a wrap around file.

Configuring a SOLA Container

Container Pre-Reqs

A SOLA CICS Container will need the following features enabled:

- CICS Web Support
- CICS Sockets (for outbound requests)
- ICSF support (for encryption)
- CICS Linkable Bridge support (for 3270 transactions)

CICS Web Support

```
Reference Manual  : CICS Internet Guide
        Sections  : 2.1  Configuring CICS Web support base components
                    2.4  Resource definition for CICS Web support
                    2.5  Resource definition for CICS Web support

Reference Manual  : CICS Installation Guide
        Sections  : 5.1 Configure CICS Web support
```

Configuration Steps:

- Define TCP port that you want to use for CICS Web support and reserve the port in the TCPIP Profile dataset (as described in the IBM Manual z/OS Communications Server: IP Configuration Reference)
  Example:

  ```
  1443 TCP TORCICS             ; CICS WEB SUPPORT
  ```

- Define CICS resource definition group SOAWEB with the following RDO definitions

  ```
  DEFINE TSMODEL(SOAWEB) GROUP(SOAWEB)
  DESCRIPTION(CICS WEB INERFACE TSQ)
  PREFIX(SOLA) LOCATION(AUXILIARY) RECOVERY(NO) SECURITY(NO)

  DEFINE TCPIPSERVICE(SOACICS) GROUP(SOAWEB)
  URM(DFHWBAAX) PORTNUMBER(<TCP Port>) STATUS(OPEN) PROTOCOL(HTTP)
  TRANSACTION(CWXN) BACKLOG(5) TSQPREFIX(SOLA) SOCKETCLOSE(10)
  MAXDATALEN(5032) SSL(NO) AUTHENTICATE(NO) GRPCRITICAL(NO)
  ```

- Include the CICS resource definition groups, DFHWEB & SOAWEB, in the group list referenced by the GRPLIST system initialization parameter of your CICS system.

- The system initialization parameters needed for CICS Web support are:

  ```
  TCPIP=YES
  ```

  The following system initialization parameters are relevant for CICS web support but can be left to default values

  ```
  DOCCODEPAGE=037         Default host code page
  LOCALCCSID=037
  WEBDELAY=(5,60)         Web timer values
  ```

- Verify CICS Web Support using supplied sample by specifying the following URL on a web browser

Example:

http://10.5.32.99:1443/CICS/CWBA/DFH$WB1A

[or]

http://mainframe.host.com:1443/CICS/CWBA/DFH$WB1A

This sample invocation should show the following text on the browser:

```
DFH$WB1A on system <CICS Region> successfully invoked through the CICS
Web Support.
```

CICS Sockets Support

```
Reference Manual   : z/OS Communications Server : IP CICS Sockets Guide
        Sections   : Chapter 2. Setting up and configuring CICS TCP/IP
```

Configuration Steps:

- Define TCP port for the CICS Sockets listener and reserve the port in TCPIP Profile dataset (as described in manual z/OS Communications Server: IP Configuration Reference)
  Example:
  ```
  1640 TCP TORCICS              ; CICS Sockets Support
  ```

- Modify CICS Startup JCL as follows
  - Add <tcphlq>.SEZALOAD to the STEPLIB concatenation
  - Add <tcphlq>.SEZATCP to the DFHRPL concatenation

- Add TCPDATA & SYSTCPD DD definitions

```
//STEPLIB   DD DISP=SHR,DSN=<cicshlq>.SDFHAUTH
//          ....|
//          DD DISP=SHR,DSN=<tcphlq>.SEZALOAD
//DFHRPL    DD DISP=SHR,DSN=<cicshlq>.SDFHLOAD
//          DD DISP=SHR,DSN=<tcphlq>.SEZATCP
//          DD DISP=SHR,DSN=SYS1.CSSLIB
//          ......
.........................
//TCPDATA   DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//SYSTCPD   DD DSN=<tcphlq>.SEZAINST(TCPDATA),DISP=SHR
```

Define/Customize EZACONFG VSAM Dataset using the following JCL.

```
//**********************************************************//
//*   THE FOLLOWING JOB DEFINES AND THEN LOADS THE VSAM    *//
//*   FILE USED FOR CICS/TCP CONFIGURATION. THE JOBSTREAM  *//
//*   CONSISTS OF THE FOLLOWING STEPS.                     *//
//*    1). DELETE A CONFIGURATION FILE IF ONE EXISTS       *//
//*    2). DEFINE THE CONFIGURATION FILE TO VSAM           *//
//*    3). ASSEMBLE THE INITIALIZATION PROGRAM             *//
//*    4). LINK THE INITIALIZATION PROGRAM                 *//
//*    5). EXECUTE THE INITIALIZATION PROGRAM TO LOAD THE  *//
//*        FILE                                            *//
//**********************************************************//
//CONFIG    JOB  MSGLEVEL=(1,1)
//*
//* THIS STEP DELETES AN OLD COPY OF THE FILE
//* IF ONE IS THERE.
//*
//DEL    EXEC  PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
   DELETE -
      <CICS TCP EZACONFG DATASET> -
      PURGE -
      ERASE
//*
//*  THIS STEP DEFINES THE NEW FILE
//*
//DEFILE EXEC  PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER (NAME(<CICS TCP EZACONFG DATASET>) VOLUMES(<CICSVOL>) -
      CYL(1 1) -
      IMBED -
      RECORDSIZE(150 150) FREESPACE(0 15) -
      INDEXED ) -
      DATA ( -
        NAME(<CICS TCP EZACONFG DATASET>.DATA) -
        KEYS (16 0) ) -
      INDEX ( -
        NAME(<CICS TCP EZACONFG DATASET>.INDEX) )
 /*
//*
//* THIS STEP ASSEMBLES THE INITIALIZATION PROGRAM
//*
//PRGDEF  EXEC PGM=ASMA90,PARM='OBJECT,TERM',REGION=1024K
//SYSLIB     DD DISP=SHR,DSNAME=SYS1.MACLIB
//             DD DISP=SHR,DSNAME=TCPIP.SEZACMAC
//SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2     DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT3     DD UNIT=SYSDA,SPACE=(CYL,(2,1))
```

```
//SYSPUNCH  DD DISP=SHR,DSNAME=NULLFILE
//SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(400,(500,50)),
//             DCB=(RECFM=FB,BLKSIZE=400,LRECL=80)
//SYSTERM   DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
         EZACICD TYPE=INITIAL,   Start of macro assembly input         X
               FILNAME=EZACICDF, DD name for configuration file        X
               PRGNAME=EZACICDF  Name of batch program to run
         EZACICD TYPE=CICS,      CICS record definition                X
               APPLID=<CICSAPPL>,  APPLID of CICS region               X
               TCPADDR=TCPIP,    Job/Step name for TCP/IP              X
               NTASKS=20,        Number of subtasks                    X
               DPRTY=0,          Subtask dispatch priority difference  X
               CACHMIN=15,       Minimum refresh time for cache        X
               CACHMAX=30,       Maximum refresh time for cache        X
               CACHRES=10,       Maximum number of resident resolvers  X
               ERRORTD=CSMT,     Transient data queue for error msgs   X
               SMSGSUP=NO        STARTED Messages Suppressed?
         EZACICD TYPE=LISTENER,  Listener record definition            X
               FORMAT=STANDARD,  Standard Listener                     X
               APPLID=<CICSAPPL>,  APPLID of CICS region               X
               TRANID=CSKL,      Transaction name for Listener         X
               PORT=<Port>,      Port number for Listener              X
               IMMED=YES,        Listener starts up at initialization? X
               BACKLOG=20,       Backlog value for Listener            X
               NUMSOCK=50,       # of sockets supported by Listener    X
               MINMSGL=4,        Minimum input message length
         EZACICD TYPE=FINAL      End of assembly input
/*
//*
//* THIS STEP LINKS THE INITIALIZATION PROGRAM
//*
//LINK    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
//             REGION=512K,COND=(4,LT)
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
//SYSLMOD   DD DSNAME=&&LOADSET(EZACICDF),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(TRK,(1,1,1)),
//             DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760)
//SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS)
  NAME EZACICDF(R)
//*
//* THIS STEP EXECUTES THE INITIALIZATION PROGRAM
//*
//FILELOAD EXEC PGM=EZACICDF,COND=(4,LT)
//STEPLIB  DD  DSN=&&LOADSET,DISP=(MOD,PASS)
//EZACONFG DD  DSNAME=<CICS TCP EZACONFG DATASET>,DISP=OLD
```

Once the EZACONFG has been setup you can use the transaction ***EZAC*** *to view/add/delete/modify the configuration file in the future.*

- Define CICS RDO definitions (Group SOCKETS) supplied in <tcphlq>.SEZAINST(EZACICCT).

  Customize the following items in the definition

  - Change the command that specifies the list to which you want the group SOCKETS to be added
    ADD GROUP(SOCKETS) LIST(your-list)
  - Change the DEFINE FILE(EZACONFG) to fill in the EZACONFG dataset

that is allocated in step before
DSNAME(<EZACONFG dataset>)

- Starting and stopping CICS sockets

  It is recommended to setup PLT Startup(PLTPI) & Shutdown(PLTSD) definitions to start/stop CICS sockets interface automatically

  To start the CICS Sockets interface automatically, make the following entry in PLTPI after the DFHDELIM entry:

  ```
  DFHPLT TYPE=ENTRY,PROGRAM=EZACIC20
  ```

  To shut down CICS Sockets interface automatically, make the following entry in the PLTSD before the DFHDELIM entry:

  ```
  DFHPLT TYPE=ENTRY,PROGRAM=EZACIC20
  ```

  If you want to control the CICS socket interface manually then use the transaction ***EZAO*** with options with options START/STOP.

- When CICS Sockets interface successfully starts you will see the following message in the JESMSGLG of CICS region.

  ```
  EZY1224I MM/DD/YY hh:mm:ss CICS/SOCKETS INITIALIZATION SUCCESSFUL USING
  REUSABLE MVS SUBTASKS
  ```

ICSF

To perform cryptographic services SOLA uses APIs provided by the Integrated Cryptographic Service Facility (ICSF). ICSF must be active and accessible for features such as XML Encryption and Decryption and XML Signature creation and validation.

- SOLA uses the cryptographic service APIs provided by ICSF
- ICSF works with the hardware cryptographic features, and is dependent on the hardware cryptographic co-processor being enabled on the S/390 server
- Ensure hardware setup is done to enable the cryptographic co-processor

Please refer to the following manual for information on installing, configuring and managing ICSF:

```
Reference Manual  : z/OS V1Rn.0 ICSF Systems Programmer’s Guide
        Sections  : 2.0  Installation, Initialization, & Customization
                    2.1  Steps for installation and initialization
                    2.5  Customizing ICSF after the first start
                    C.1  Checklist for First-Time Startup of ICSF
```

ICSF Configuration Summary:

- Verify the following changes to your installation parmlib:
  - Add CSF.SCSFMOD0 to the LNKLST concatenation
  - Update PROGxx to APF authorize CSF.SCSFMOD0
  - Update IKJTSOxx for ICSF by adding CSFDAUTH and CSFDPKDS to the AUTHPGM and AUTHTSF parameter lists

- Allocate an empty Cryptographic Key Data Set (CKDS) using the sample JCL in SYS1.SAMPLIB(CSFCKDS)

- Allocate an empty Public Key Data Set (PKDS) using the sample JCL in SYS1.SAMPLIB(CSFCKDS)

- Copy IBM supplied sample parmlib from SYS1.SAMPLIB(CSFPRM00) to the installation parmlib dataset & customize parms CKDSN(<allocated CKDS DSN>) & PKDSN(<allocated PKDS DSN>).

- Copy IBM supplied STC proclib SYS1.SAMPLIB(CSF) to the installation proclib dataset & customize the following

  ```
  //CSFPARM DD DSN=<Installation Parmlib Dataset>(CSFPRM00),DISP=SHR
  ```

- Setup RACF definitions for defining the userid with which the started task **CSF** should run (RACF Class : STARTED)

- Start the CSF started task from console (/S CSF). You will see error messages in the SYSLOG about CKDS/PKDS not being initialized & Master keys not being valid. Apart from this you will also see messages:
  CSFM001I ICSF INITIALIZATION COMPLETE
  CSFM400I CRYPTOGRAPHY - SERVICES ARE NOW AVAILABLE.

- Setup the ICSF ISPF interface by copying the following invocation exec into your local shared rexx exec library that is concatenated to the SYSEXEC DD.

```
/* Rexx */
/* IBMs ICSF */
ADDRESS ISPEXEC
"LIBDEF ISPPLIB DATASET ID('CSF.SCSFPNL0') STACK"
"LIBDEF ISPMLIB DATASET ID('CSF.SCSFMSG0') STACK"
"LIBDEF ISPSLIB DATASET ID('CSF.SCSFSKL0') STACK"
"LIBDEF ISPTLIB DATASET ID('CSF.SCSFTLIB') STACK"
ADDRESS TSO "ALTLIB ACTIVATE APPLICATION(CLIST)
           DATASET('CSF.SCSFCLI0')"
"SELECT PANEL(CSF@PRIM)"
ADDRESS TSO "ALTLIB DEACTIVATE APPLICATION(CLIST)"
"LIBDEF ISPSLIB"
"LIBDEF ISPPLIB"
"LIBDEF ISPMLIB"
"LIBDEF ISPTLIB"
```

Optionally you can integrate this exec into your local ISPF or Site menu.

- Invoke ICSF ISPF interface and choose option :

6 PPINIT - Pass Phrase Master Key/CKDS Initialization

Enter the pass phrase and CKDS/PKDS DS names to initialize & setup PKA master keys

```
------------------- ICSF - Pass Phrase MK/KDS Initialization -------
COMMAND ===>

 Enter your pass phrase and the names of the CKDS and PKDS:

   Pass Phrase (16 to 64 characters)
   ===> <Pass Phrase>
   CKDS
   ===> <CKDS DSN>
   PKDS
   ===> <PKDS DSN>

   Initialize the CKDS and PKDS? (Y/N) ===> Y
   Signature MK = Key Management MK? (Y/N) ===> Y
   Initialize new PCICCs only ? (Y/N) ===> N



Press ENTER to process.
Press END   to exit to the previous menu.
```

Recycle the CSF Started task and verify if that there are no error messages

CICS LINK3270 (3270 Bridge) Support

```
Reference Manual : CICS External Interfaces Guide
        Sections : Part 2. Bridging to 3270 transactions (Chapters 2-7)
```

The 3270 bridge provides an interface so that you can run 3270-based CICS transactions without a 3270 terminal. The 3270 terminal and end-user are replaced by an application program, known as the client application. Commands for the 3270 terminal in the CICS 3270 user transaction are intercepted by CICS and replaced by a messaging mechanism that provides a bridge between the client application and the CICS user transaction.

SOLA exploits Link3270 feature(DFHL3270) in session mode for BMS support

Configuration Steps:

- Define a DFHBRNSF file: The bridge facility name space allocation file (DFHBRNSF) can be a local user data table, a local VSAM file, a coupling facility data table (CFDT), a remote VSAM file or a VSAM RLS file depending on your configuration needs. The following VSAM File and FCT entry definitions reflect a local empty VSAM file DFHBRNSF file.

```
//DEFDS JOB accounting info,name,MSGCLASS=A
//DFHBRNSF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
      DEFINE CLUSTER(NAME(<HLQ>.SOLA.DFHBRNSF)-
            INDEXED-
            TRK(10 10)-
            RECORDSIZE(384 384)-
            KEYS(13 20)-
            FREESPACE(0 50)-
            SHAREOPTIONS(2 3)-
            LOG(NONE)-
            VOLUME(DISK01)-
            CISZ(512)) -
      DATA (NAME(<HLQ>.CICS.DFHBRNSF.DATA)-
            CISZ(512)) -
      INDEX (NAME(<HLQ>.CICS.DFHBRNSF.INDEX)-
            CISZ(512))
/*
```

The FCT definition given below corresponds to a LOCAL DFHBRNSF Non-RLS file definition and has to be defined to the CICS regions where SOLA Run-time environment is setup

```
DEFINE FILE(DFHBRNSF) GROUP(<SOLAGRP>)
DESCRIPTION(3270)
       DSNAME(<HLQ>.SOLA.DFHBRNSF) RLSACCESS(NO) LSRPOOLID(1)
```

```
            READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS) STRINGS(1)
            STATUS(ENABLED) OPENTIME(STARTUP) DISPOSITION(SHARE)
            DATABUFFERS(2) INDEXBUFFERS(1) TABLE(NO)
            MAXNUMRECS(NOLIMIT)
            UPDATEMODEL(LOCKING) LOAD(NO) RECORDFORMAT(V) ADD(YES)
            BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES) JOURNAL(NO)
            JNLREAD(NONE) JNLSYNCREAD(NO) JNLUPDATE(NO) JNLADD(NONE)
            JNLSYNCWRITE(YES) RECOVERY(NONE) FWDRECOVLOG(NO)
            BACKUPTYPE(STATIC)
```

- DFHSIT Customization
    - AIBRIDGE=AUTO

      This parameter specifies that the autoinstall URM is not called when bridge facilities are created and deleted.

    - BRMAXKEEPTIME=28800

      The default maximum timeout value (24 hours) that a client can specify to retain an unused bridge facility before it is deleted.

- Defining the bridge facility

  The bridge facility is an emulated 3270 terminal. It is a virtual terminal, created by DFHL3270 when it receives a single transaction mode request, or a session mode request to allocate a bridge facility. You do not provide a TERMINAL resource definition for the bridge facility, but you can control the terminal properties used by providing a 3270 TERMINAL resource definition to be used as a template. This TERMINAL definition, is known as the facilitylike.

  It is recommended to use the default terminal facilitylike definition **CBRF** that is supplied by CICS in **DFHTERM** Group.

```
DEFINE TERMINAL(CBRF) GROUP(DFHTERM)
DESCRIPTION(Default 3270 Bridge template)
        AUTINSTMODEL(NO) TYPETERM(DFHLU2) NETNAME(CBRF)
        REMOTESYSTEM(CBR) REMOTENAME(CBRF) PRINTERCOPY(NO)
        ALTPRINTCOPY(NO) TASKLIMIT(NO) TERMPRIORITY(0)
        INSERVICE(YES) SOLICITED(NO) ATTACHSEC(LOCAL)
        BINDSECURITY(NO) USEDFLTUSER(NO)
```

- Verify if the CICS Supplied Resource definition Group **DFHBR** is installed in the region. This resource group will install 3 CICS PPT definitions for the programs:
    - DFHBRCV
    - DFHBRMP
    - DFHL3270

Managing LINK3270:

- Type `CEMT INQUIRE BRFACILITY` (the minimum abbreviation is `CEMT I BR`) to get a display that lists the status of any currently allocated bridge facilities.
- Type `CEMT INQUIRE TASK BRFACILITY` (the minimum abbreviation is `CEMT I TA BR`) to get a display of tasks with a 8-byte field containing the facilitytoken for the bridge facility in use by the task.
- Use `CECI {SET / INQUIRE} TRACETYPE` to enable / inquire bridge tracing

AT-TLS

Background

Application Transparent Transport Layer Security (AT-TLS) creates a secure session on behalf of an application. Instead of implementing SSL/TLS in SOLA for a secure outbound connection, AT-TLS will provide encryption and decryption of data based on minimal set of policy statements coded & enabled in the Policy Agent running on host. The application sends and receives cleartext (unencrypted data) as usual while AT-TLS encrypts and decrypts data at the TCP transport layer.

The advantage of exploiting this feature that has been enabled since z/OS V1.7 is that we can easily enable support for TLS V1 and in future when IBM enables support for newer versions of TLS we automatically support the new versions. Of course a customer needs to be at a level of z/OS V1.7 or higher for SOLA to exploit this feature.

AT-TLS Setup

```
Reference Manuals :
      z/OS V1Rn.0 CS:IP Configuration Guide
      Chapter 18. Application Transparent Transport Layer Security (AT-TLS)
      data protection

      z/OS V1Rn.0 CS:IP Configuration Reference
      Sections  : Chapter 21. Policy Agent and policy applications
```

- Enable “TCPCONFIG TTLS” in the “PROFILE” member of the TCPPARMS. To dynamically activate the new profile issue MVS Vary command
  VARY TCPIP,,CMD=OBEYFILE,DSN=<TCPPARMSDS>(<ProfileMember>)

- Define a Keyring in RACF for each of the following userids
  - For batch applications the Keyring needs to be created for the userid under which the batch job runs

- For CICS regions the Keyring needs to be created for the Userid under which CICS runs (Not the userid under which txn runs)

  ```
  RACDCERT ADDRING(APPKEYRING) ID(<ApplicationOwner>/<CICS ID>)
  ```

- Get the CA Root certificates of all the servers to which you will be connecting. Connect the CA Root certificates to the keyrings. These certificates are used for validating Server certificate during SSL handshake. Alternately you could import the server certificates directly into RACF with Trust status and connect it to the keyrings to bypass certificate validation.

  Import & Connect Server CA Root certificate to the keyring

  ```
  RACDCERT CERTAUTH ADD('<Dataset with CA Root Cert>')
  WITHLABEL('Server CA')

  RACDCERT ID(<CICS ID>)     CONNECT(CERTAUTH LABEL('Server CA')
  RING(APPKEYRING) USAGE(CERTAUTH))
  ```

- ** Optional Step** In case Server is configured for ClientAuth & expects to validate client certificate **

  Create a certificate for usage by the SOLA for outgoing webservice calls. If you have a local Certifying Authority(CA) then export the certificate to get it signed by CA & re-import the signed certificate back in RACF.

  Example

  - Add a site certificate for usage by SOLA (Can be done through RACF Administration Panels or with use of RACDCERT Command)

    ```
    RACDCERT SITE GENCERT +
     SUBJECTSDN( CN('SOLA Client Cert') OU('SOA') C('US') ) +
     WITHLABEL('SOLA Prod Client Cert') SIZE(1024)
    ```

  - Generate a PKCS10 Base64 encoded certificate request. From RACF Administration panel use option 7 (DIGITAL CERTIFICATES AND KEY RINGS), subtoption 2 (Create a certificate request).

    Type "S" on Certificate type "Site", give the Certificate Label as defined in the earlier step ('SOLA Prod Client Cert'), specify a dataset into which the PKCS10 certificate request is to be generated. RACF will automatically allocate the dataset.

  - Xfer the PKCS10 Certificate request (alternatively you can cut & paste the BASE64 contents in the datasets) to a file that can be sent to the security team for getting it signed by a CA.

  - Make sure you get your signed certificate as well as the CA root certificate & Import the certificates into RACF with appropriate Certificate Type & labels

(You can use RACDCERT command or option 7;4;1(RACF - Add Digital Certificate). Make sure you add the certificates with Trust status.

```
RACDCERT CERTAUTH ADD('<Dataset having Local CA Cert>')
WITHLABEL('Local CA')

RACDCERT ID(<USERID>) ADD('<Dataset having Singed Cert') -
WITHLABEL('SOLA Prod Client Cert')
```

- Connect Signed certificate & the CA certificate to the Keyrings created in earlier step (Can be done through RACF panels or using RACDCERT Command). Make sure the Signed certificate is connected as DEFAULT certificate of the Keyring.

  Example:
  Connect CA certificate to the keyring

  ```
  RACDCERT ID(<CICS ID>)     CONNECT(CERTAUTH LABEL('LOCAL CA')
  RING(APPKEYRING) USAGE(CERTAUTH))
  ```

  Connect signed certificate to the keyring

  ```
  RACDCERT ID(<CICS ID>) CONNECT(SITE LABEL('SOLA Prod Client Cert')
  RING(APPKEYRING) USAGE(PERSONAL) DEFAULT)
  ```

- RACF Definitions needed for accessing certificates

```
RDEFINE FACILITY  (IRR.DIGTCERT.LISTRING )  UACC(NONE)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)
   ID(<Appl user>/<CICS user>) ACCESS(CONTROL)

RDEFINE FACILITY  (IRR.DIGTCERT.GENCERT )  UACC(NONE)
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY)
   ID(<Appl user>/<CICS user>) ACCESS(CONTROL)
```

- Define “Policy Configuration” & “Policy TTLS” configuration members. Sample members can be found in Appendix C: AT-TLS Sample Configuration Data on page 209.

- Define Policy Agent Started Task (PAGENT) and environment setup. Sample members can be found in Appendix C: AT-TLS Sample Configuration Data on page 209.

- Start PAGENT STC

- Add the startup/shutdown of PAGENT STC into your site automation rules. Alternately the PAGENT can be integrated and started as a part of AUTOLOG process in TCPIP PROFILE. TCPIP will wait for PAGENT to come up before allowing any other ports to be opened by FTP, TELNET daemons. To allow

PAGENT to open the port the id with which PAGENT will run should have READ access to the RACF Class SERVAUTH with Name EZB.**.

Configuring an AOR Region for SOLA

When to Configure an AOR

This chapter details the steps necessary to configure an AOR container for use with SOLA.

It is only necessary to configure an AOR under certain conditions. Some of the configuration steps for certain conditions may conflict with the configuration required for other conditions , therefore it is recommended that the AOR be configured on an as needed basis for the required conditions only.

At a glance:

Plugin Type	AOR Components	RDO for SOLA Components
Commarea	Not Required	N/A
Callable	Required	PPT
BMS3270*	Required only for test containers (analysis)	Test containers only: PPT, PCT,RCT
Custom	Required	PPT
Dynamic SQL	Required	PPT (PCT, RCT)
Stored Procedure	Required	PPT (PCT, RCT)
Outbound Plugin	Required	PPT

*** See the SOLA 3270 Plugin section below for more specific instructions.**

SOLA™ NOTE

Additional TOR configuration may also be necessary under some circumstances. The required configuration steps are detailed in this section.

How to Configure an AOR

SOLA 3270 Plug-in

SOLA uses the CICS bridge facility. If you are using the linkable bridge (recommended for CICS 2.2 and above), it must be defined in both the TOR and AOR (see IBM CICS manuals for information). If you are not using the linkable bridge, then you do not need to define the bridge facility in the TOR (still required in AOR).

Additionally, the CICS linkable bridge requires a terminal CBRF to be defined in all AOR containers in order to execute transactions over the bridge facility (detailed below).

SOLA-specific AOR configuration is not required for running 3270 transactions as web-services using the CICS linkable bridge. However, in order to create those web-services (Analysis), SOLA requires that some of its components be in the AOR containers.

The following entries must be present in the test AOR containers that will be used during the analysis (where user transactions would run):

- **Local PPT definitions:**
  - XMLPC101
  - XMLPC110
  - XMLPC201
  - XMLPC400
  - XMLPC403
  - XMLPC404
  - XMLPC406.

- **Local PCT definition:** for XML# transaction, as follows:

```
DEFINE TRANSACTION(XML#)
GROUP(<group>)            PROGRAM(DFHMIRS)              TWASIZE(0)
 PROFILE(DFHCICSA)        STATUS(ENABLED)
 TASKDATALOC(BELOW)       TASKDATAKEY(USER)       STORAGECLEAR(NO)
 RUNAWAY(SYSTEM)          SHUTDOWN(DISABLED)      ISOLATE(YES)
 PRIORITY(100)            TRANCLASS(DFHTCL00)
 DTIMOUT(NO)              RESTART(NO)             SPURGE(NO)
 TPURGE(NO)               DUMP(YES)               TRACE(YES)
 CONFDATA(NO)             OTSTIMEOUT(NO)
 ACTION(BACKOUT)          WAIT(YES)               WAITTIME(0,0,0)
RESSEC(NO)               CMDSEC(NO)
```

Please note that CSD Definition and XML# definition for TRANCLASS is shipped with the default DFHTCL00 and must be customized at setup.

- **Local RCT definition:** for XML# transaction pointing to XMLPLAN.

```
DEFINE DB2ENTRY(XML#)
GROUP(<group>)             TRANSID(XML#)
ACCOUNTREC(TXID)           AUTHID(<authorized id>)
DROLLBACK(YES)             PLAN(<plan>)
PRIORITY(LOW)              PROTECTNUM(0)          THREADLIMIT(0)
THREADWAIT(POOL)
```

Please note that AUTHID must be have DB2 SYSADM authority or an authorization id that will be passed via SOAP request.

Please note that the above definitions are only needed in the test container to enable the analysis. Once the analysis is completed, SOLA doesn’t require any of its components in the AOR container. In the production or QA environment, no SOLA-specific AOR configuration is required (CICS configuration may still be needed).

Below is a sample definition for the CBRF terminal that must be present in all the AOR containers to successfully run a transaction using CICS bridge facility.

```
DEFINE                 TERMINAL(CBRF)         GROUP(<group>)
DESCRIPTION(DEFAULT 3270 BRIDGE TEMPLATE)
TYPETERM(DFHLU2)       NETNAME(CBRF)          CONSOLE(NO)
ATTACHSEC(LOCAL)       BINDSECURITY(NO)       USEDFLTUSER(NO)
TYPETERM(DFHLU2)       DEVICE(LUTYPE2)        TERMMODEL(2)
PAGESIZE(24,80)        ALTPAGE(0,0)           FMHPARM(NO)
DEFSCREEN(24,80)       ALTSCREEN(   ,   )     APLKYBD(NO)
APLTEXT(NO)            AUDIBLEALARM(YES)      COLOR(NO)
COPY(NO)               DUALCASEKYBD(NO)       EXTENDEDDS(YES)
HILIGHT(NO)            KATAKANA(NO)           LIGHTPEN(NO)
MSRCONTROL(NO)         OBFORMAT(NO)           PARTITIONS(NO)
PRINTADAPTER(NO)       PROGSYMBOLS(NO)        VALIDATION(NO)
FORMFEED(NO)           HORIZFORM(NO)          VERTICALFORM(NO)
TEXTKYBD(NO)           TEXTPRINT(NO)          QUERY(ALL)
OUTLINE(NO)            SOSI(NO)               BACKTRANS(NO)
CGCSGID(0,0)           SENDSIZE(1536)         RECEIVESIZE(256)
BRACKET(YES)           ERRINTENSIFY(YES)      ERRCOLOR(NO)
ERRLASTLINE(YES)       ERRHILIGHT(NO)
AUTOCONNECT(NO)        ATI(YES)               TTI(YES)
CREATESESS(NO)         RELREQ(YES)            DISCREQ(YES)
NEPCLASS(0)            SIGNOFF(YES)           RSTSIGNOFF(NOFORCE)
ROUTEDMSGS(ALL)        LOGONMSG(YES)
BUILDCHAIN(YES)        USERAREALEN(0)         IOAREALEN(256,4000)
UCTRAN(YES)            RECOVOPTION(SYSDEFAULT)
```

SOLA Custom Program Plugin

The following programs need LOCAL PPT entries defined in the AOR container to run a custom plugin:

- **XMLPC101**
- **XMLPC110**
- **XMLPC200**
- **XMLPC201 (Custom Plugin using Old DOM API XMLPC110)**
- **XMLPC202 (Custom Plugin using new DOMA PI XMLPC112)**

SOLA Callable Program Plugin

The following programs need LOCAL PPT entries defined in the AOR container for SOLA's Callable plugin:

- **XMLPC101**
- **XMLPC110**
- **XMLPC201**
- **XMLPC205 (Doesn't pass DFHEIBLK Parameter)**
- **XMLPC206 (Pass DFHEIBLK parameter)**

SOLA Outbound Plugin

See section Outbound Plugin AOR Configuration on page 124.

SOLA Dynamic SQL Plugin

When executing dynamic SQL as a web-service, SOLA's components run in the AOR container that executes the SQL and returns the data as soap response via the TOR.

The following local PPT entries are needed in the AOR:

- **XMLPC101**
- **XMLPC110**
- **XMLPC200**
- **XMLPC201**
- **XMLPC251**

The following local PCT entries are needed in the AOR:

- Any mirror transaction (e.g. XML#)

The following local RCT entries are needed in the AOR:

- XML# transaction (or any other mirror transaction) needs to use a DB2 Plan that can execute dynamic SQL. Select from one of the following options:

    a. Use XMLPLAN and GRANT sufficient auth to XMLPLAN for executing the dynamic SQL.

    b. Use your own plan that already has authority and add XML.* in the plan's collection list.

TOR Configuration (to activate dynamic SQL plugin)

Create a new PPT definition for programs that have been registered (in the SOLA IDE) to execute dynamic SQL.

```
DEFINE PROGRAM(registered program name) GROUP(<group>)
LANGUAGE(LE370) RELOAD(NO)
RESIDENT(NO) USAGE(NORMAL)   USELPACOPY(NO)
STATUS(ENABLED) CEDF(YES) DATALOCATION(BELOW)
REMOTESYSTEM(<AOR sysid>) REMOTENAME(XMLPC200)  TRANSID(XML# or any
other chosen mirror tranid)
```

Please note that remote name and program name are intentionally different.

SOLA Stored Procedure Plugin

When executing a stored procedure as a web-service, SOLA's components run in the AOR container that invokes the stored procedure and returns the data as soap response via the TOR.

The following local PPT entries are needed in AOR:

- **XMLPC101**
- **XMLPC110**
- **XMLPC200**
- **XMLPC201**
- **XMLPC250**

The following local PCT entries are needed in the AOR:

- Any mirror transaction (e.g. XML#)

The following local RCT entries are needed in the AOR:

- XML# transaction (or any other mirror transaction) needs to use a DB2 Plan that can execute a stored procedure. Select from one of the following options:

  c. Use XMLPLAN and GRANT sufficient auth to XMLPLAN for executing the stored procedure.

  d. Use your own plan that already has authority and add XML.* in the plan’s collection list.

TOR Configuration (to activate stored procedure plugin)

Create a new PPT definition for programs that have been registered (in the SOLA IDE) to execute stored procedures.

```
DEFINE PROGRAM(registered program name)  GROUP(<group>)
LANGUAGE(LE370) RELOAD(NO)
RESIDENT(NO) USAGE(NORMAL)   USELPACOPY(NO)
STATUS(ENABLED) CEDF(YES) DATALOCATION(BELOW)
REMOTESYSTEM(<AOR sysid>) REMOTENAME(XMLPC200)  TRANSID(XML# or any
other chosen mirror tranid)
```

Please note that remote name and program name are intentionally different.

SOLA™	NOTE	We recommend that a *.* collection be added to XMLPLAN in the test container and the authid that runs XML# transaction should be granted enough authorization (for example SYSADM) to run any stored procedure.

Configuring the SOLA Outbound Plugin

Outbound Plugin AOR Configuration

The following remote PPT definition is needed in the AOR containers. The definition should point to the TOR container where SOLA is installed.

```
DEFINE PROGRAM(XMLPC103) GROUP(<group>)
LANGUAGE(LE370) RELOAD(NO)
RESIDENT(NO) USAGE(NORMAL)   USELPACOPY(NO)
STATUS(ENABLED) CEDF(YES) DATALOCATION(BELOW)
REMOTESYSTEM(<TOR sysid>) TRANSID(XML#)
```

Outbound Plugin TOR Configuration

The following are required to activate the outbound plugin.

The TOR container (where SOLA is installed) must have an RCT entry defined for XML# pointing to XMLPLAN.

To run the outbound plugin in CICS, CICS Sockets must be installed in the TOR container. This is not required to run outbound plugin in batch mode.

Configuring the SOLA IMS Container

The SOLA IMS Container (referenced in this section as SOLA STC) runs as a z/OS started task. It incorporates many of the features of the SOLA CICS Container, but because it runs without CICS there are several operational parameters that it requires.

The parameters are specified in a dataset referenced by the PARMLIB DD statement in the Started Task JCL.

Started Task JCL

The following sample JCL is provided in the SOLA SAMPLIB. Customize this JCL to conform to your installation requirements. The JCL can be customized using the SOLAEDT Rexx Edit Macro that you customized during the installation of SOLA.

SOLA STC Proclib

```
//*  SOLA STARTED TASK
//LISTEN  EXEC PGM=XMLPC125,
//            REGION=<RegionSize>,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=<tlibhlq>.LOADLIB
//         DD DISP=SHR,DSN=<db2SDSNLOAD>
//SOLALIB  DD DISP=SHR,DSN=<tlibhlq>.LOADLIB
//         DD DISP=SHR,DSN=<Application Template Libary>
//SYSUT1   DD DSN=&UT1,
//            SPACE=(1700,(400,50)),
//            UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*DSNTRACE DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//PARMLIB  DD DISP=SHR,DSN=<Parmlib(SOLAPRMS)>
```

<RegionSize>
The region size that’s required to run the SOLA STC can be computed as follows:

Base Region size needed = 3 MB
Each SOLA Thread will consume = 0.5 MB

If you setup a SOLA Started task to process a maximum of 50 threads then the region size required is 3 MB + (50*.0.5 MB) = 28 MB

<Application Template Library(Libraries)>
Concate your application template library(libraries) to the SOLALIB DD card. These datasets are load libaries where generated SOLA template loadmodule artefacts are stored

<Parmlib(SOLAPRMS)

```
TRCE=0
SYST=SOL1
PORT=01449
PLAN=XMLCLNT
DB2S=DB8G
PROT=000
MAXT=200
FCTM=02000
TCPN=TCPIP
TIME=00100
MTSQ=09999
IDLE=86400
```

Limited validation is performed on the parameters, so it’s imperative that you specify each parameter exactly as shown. Each parameter must start on a separate line and must begin in column 1. Don’t abbreviate numeric fields; each field must have the requisite number of digits, as specified below.

SYST: Mandatory
4 alphanumeric characters. The SOLA System ID (equivalent of CICS SYSID) assigned to the SOLA Started Task instance.

PORT: Mandatory
5 numeric digits. SOLA Started Task Listener TCPIP Port Number

DB2S: Mandatory
4 alphanumeric characters. DB2 Subsystem in which the SOLA directory is created

PLAN: Mandatory
8 alphanumeric characters. DB2 Plan to be used

TRCE: Optional
1 numeric digit. Sets the trace Level for debugging the SOLA Started Task.

Values : 0 – 9.

Default: 0 (No Trace)
Tracing is set off with value 0 and tracing is more verbose the higher the number specified. Tracing is primarily used by SOLA support, who will provide you with the appropriate value to use.

PROT: Optional
3 numeric digits. Specifies the number of protected threads to be created.

Default: 000 (No protected threads created at start-up of SOLA STC instance)

MAXT: Optional
3 numeric digits. Specifies the maximum number of concurrent threads to be supported by the SOLA STC instance

TCPN: Optional
8 alphanumeric characters. TCPIP address space name on the system that SOLA will connect to.

Default: TCPIP

FCTM: Optional
5 numeric digits. Specifies the number of SOLA Internal Logging File Control records to be handled by the SOLA STC instance.

Default: 02000

MTSQ: Optional
5 numeric digits. SOLA STC caches runtime metadata into internal memory areas called TSQs. This parameter defines the number of TSQs to be supported by the specific SOLA STC instance.

Default: 09999

TIME: Optional
5 numeric digits. This is an SOLA internal control parameter that indicates how long in milliseconds that the listener should wait before the incoming socket connection is taken by a subtask.

Default: 00100

IDLE: Optional

5 numeric digits. SOLA STC thread manager uses the value specified in this parm to control when an IDLE thread is to be released. The value specified in this parm indicates number of seconds after which an Inactive(Idle) SOLA thread needs to be terminated
Default: 86400

SOLA IMS Container Operator Commands

The SOLA IMS Container (referenced in this section as SOLA STC) runs as a z/OS started task. It incorporates many of the features of the SOLA CICS Container, but it isn't running under CICS. To allow you to control its operation. it provides a set of operator commands (and alternative web services) that you can use to

The commands are designed to be entered on the z/OS operator's Console. The alternative web service requests can be entered in the "SOAP Tester" panel in the SOLA Developer.

/S <SOLASTC>
To start SOLA Started Task instance <SOLASTCName>

/P <SOLASTC> (or) /F <SOLASTC>,STOP
To stop SOLA Started Task instance <SOLASTC>

/F <SOLASTC>,STOP
To stop SOLA Started Task instance <SOLASTC>

/F <SOLASTC>,DISP
Display the status of threads in the SOLA Started Task.

Alternatively, choose the SOLA Started Task end-point and enter the following SOAP request in the SOAP Tester:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ObjectService
xmlns="http://project.ObjectFinder.x4mlsoa.com/SL/XMLPC804/">
         <Operation>select</Operation>
         <Object objectType="STC" operationType="displayThreads"/>
      </ObjectService>
   </soap:Body>
</soap:Envelope>
```

/F <SOLASTC>,TERM,THREAD=<ThreadNbr>
To terminate a SOLA thread in the SOLA Started Task. The value <ThreadNbr> is the thread number, which you obtain from the display command above.

/F <SOLASTC>,TRACE=ON,LEVEL=<Level>
/F <SOLASTC>,TRACE=ON,PROG=<ProgramMask>
To start and configure the SOLA Started Task trace. There are multiple trace levels <Level> and a wildcard capability <ProgramMask> that control which program(s) write

trace records. Trace commands to be issued will be provided by SOLA Support. Tracing generate internal trace records that are used to diagnose problems.

/F <SOLASTC>,TRACE=OFF
To stop SOLA Started Task trace

/F <SOLASTC>,START,STATS
When the SOLA Started Task instance starts it will automatically start a special thread called the STATS thread. The function of this thread is logging, cache management and auditing of application SOAP requests processed by SOLA.

The STATS thread attaches a DB2 thread. In case of exception or abnormal termination of the STATS thread, following message is sent to <SOLAStc> address space.

'SOLA210S XMLPC122 Abended. Check job output for dump and resolve issue before restart upon console'

Typical reasons for this thread failing are DB2 subsystem failures and DB2 Resource issues.

In the event of a failure of the STATS thread, check the logs and correct the problem. Finally, issue this command to restart the STATS thread.

Contact SOA Software support in case of any dumps.

/F <SOLASTC>,HELP or /F <SOLASTC>,?
Displays a list of operator commands on the console

Refreshing Templates in the SOLA STC

The SOLA STC uses Templates to store run-time meta data. A Template is an Assembler Data-Only Load Module. For performance reasons, the SOLA STC manages the loading and caching of Templates. Ordinarily this isn't an issue, but when you change a Template you need to refresh the Template in the SOLA STC.

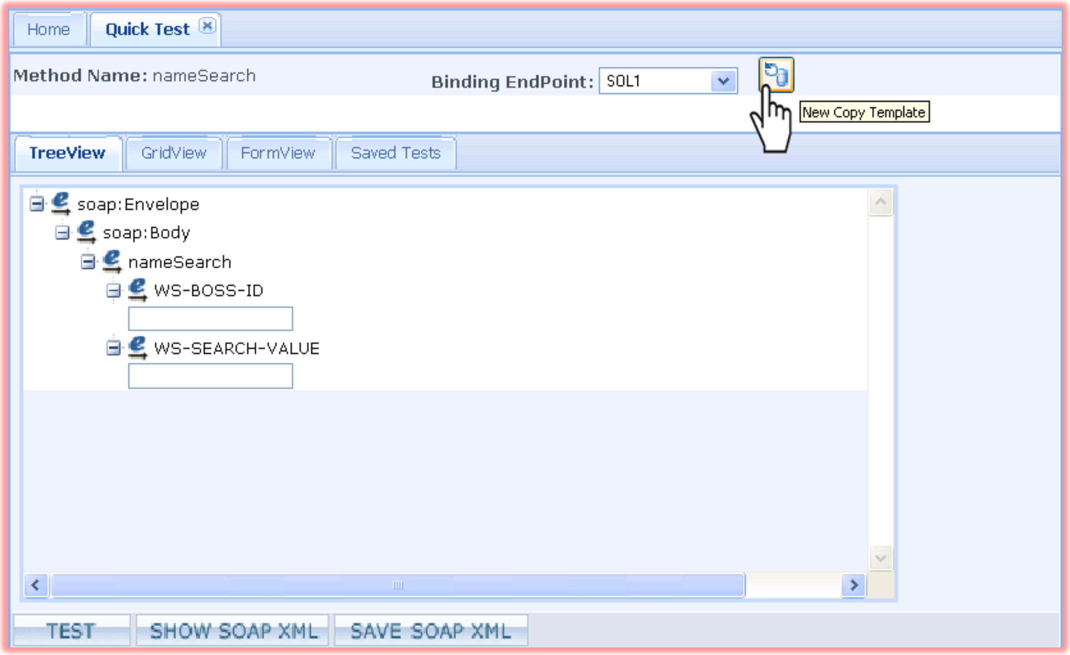
The SOLASTC provides two methods of refreshing a template:

- A manual method intended to be used by a programmer
- A web service method intended to be used for integration with a Change Management system.

Manually Refreshing a Template

CICS provides the ability to "NewCopy" a program with the CEMT transaction. SOLA IMS Container provides the same ability to NewCopy a template (the only user modifiable component hosted in the SOLA STC), but instead of providing a transaction SOLA provides a refresh button on the Quick Test pane.

After analyzing a new method, right click on the Method and choose Quick Test to bring up the Quick Test pane. In the upper right of the pane is a refresh button. Pressing this button refreshes the Template in the SOLA STC identified by the Binding Endpoint.



Method Name: nameSearch

Binding EndPoint: SOL1

```
soap:Envelope
  soap:Body
    nameSearch
      WS-BOSS-ID
      WS-SEARCH-VALUE
```

Refreshing a Template Using the Web Service Interface

SOLA provides integration points with your Change management system. One of those points is the NewCopy web service. By integrating the NewCopy service, you will be able to ensure that a template is available for use.

The following web service request can be executed against the SOLA STC to refresh the template:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ObjectService xmlns="http://project.ObjectFinder.x4mlsoa.com/SL/XMLPC804/">
         <Operation>select</Operation>
         <ObjectType>SOLAUtil</ObjectType>
         <Object objectType="SOLAUtil" operationType="newCopy"
programNm="[TemplateName]"/>
      </ObjectService>
   </soap:Body>
</soap:Envelope>
```

Replace [TemplateName] with the name of your template. The service can be executed from any web services client, including the SOLA Test Harness. Many customers use a SOLA Outbound Service from the SOLA Batch Container.

Listener Groups and Containers

Creating SOLA authorized listener groups and populating those groups with containers is done using the SOLA Resource Manager. In the case of a CICS region, the SOLA installation steps must be performed for that region. Please see the SOLA Installation Guide for details on installing SOLA software into a CICS region.

The first step in using Resource Manager is the creation of container groups and containers. For SOLA CICS Container, containers represent actual CICS TOR regions on the mainframe that SOLA will need to interact with, while container groups represent groupings of those containers. For SOLA IMS Container, which runs as a z/OS started task and incorporates many of the features of the SOLA CICS Container, containers represent the STC.

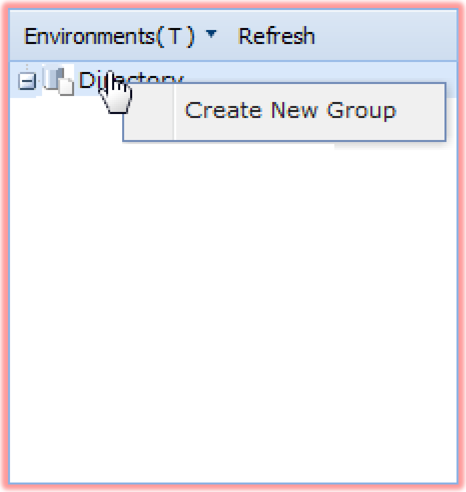
When you first launch Resource Manager, the Containers panel will be empty. It will be up to you to populate it with all of the SOLA containers you will need to use with SOLA and to organize those containers into groups.

Container Groups

Container groups are more than just containers for groups; they also serve to control metrics collection and security policies for the containers inside them. Groups allow you to do the following:

- Enable and configure metrics collection
- Enable use of the default security policy
- Designate a security user exit
- Configure cache, queue and storage options
- Enable and configure custom security policy

Creating Container Groups

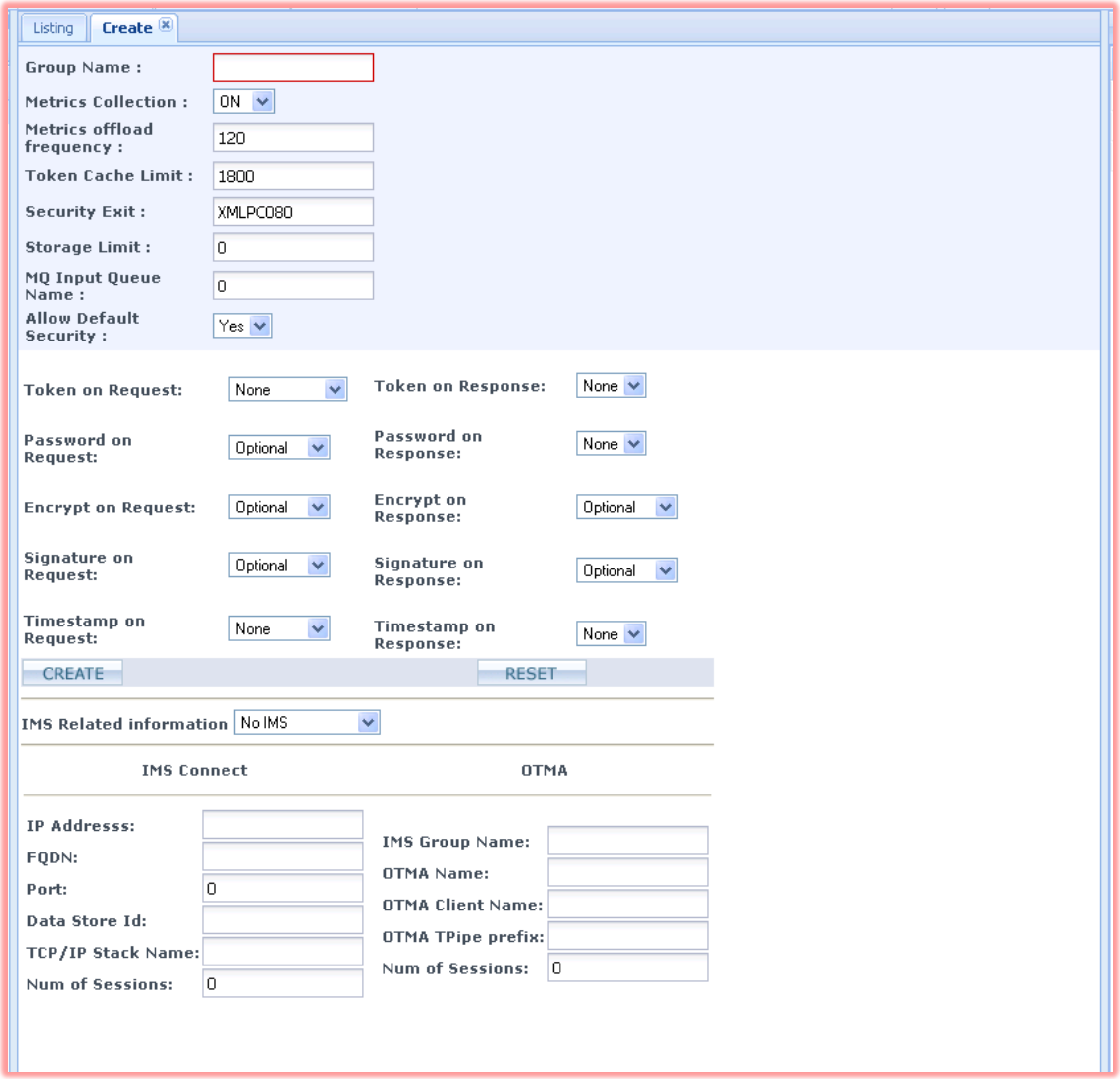


All containers in Resource Manager must be contained in a container group. To create a container group, right click on the directory icon and select **Create New Group** from the menu.

This will display the Create tab in the workspace, allowing you to create a new container group.

We recommended that you group your containers based on their security level (low security group for test containers, high security group for production containers, etc.). This will make assigning access a lot simpler.

The create tab contains a series of fields that you will need to populate to create a new group. All fields/menus except for the group name are preconfigured with the default settings.



Group Name :
Metrics Collection : ON
Metrics offload frequency : 120
Token Cache Limit : 1800
Security Exit : XMLPC080
Storage Limit : 0
MQ Input Queue Name : 0
Allow Default Security : Yes

Token on Request: None
Token on Response: None
Password on Request: Optional
Password on Response: None
Encrypt on Request: Optional
Encrypt on Response: Optional
Signature on Request: Optional
Signature on Response: Optional
Timestamp on Request: None
Timestamp on Response: None

IMS Related information No IMS

IP Addresss:
FQDN:
Port: 0
Data Store Id:
TCP/IP Stack Name:
Num of Sessions: 0

IMS Group Name:
OTMA Name:
OTMA Client Name:
OTMA TPipe prefix:
Num of Sessions: 0

To create a group with the default settings, fill in the **Group Name** field and click **CREATE**. To configure custom settings for the group, you will need to make changes to the following settings.

Standard Settings:

- **Metrics Collection:** enables (ON) or disables (OFF) metrics collection (by SOLA) for the containers in the group.
- **Metrics offload frequency:** determines how often, in seconds, metrics are spooled to the database.
- **Token Cache Limit:** How long, in seconds, before a cached token expires

- **Security Exit:** specifies the program to be used as security exit. By default, XMLPC080, the SOLA security exit, is used

- **Storage Limit:** the maximum size of an outbound message

- **MQ Input Queue Name:** the name of the MQ queue that SOLA will listen to for input MQ messages

- **Allow Default Security:** specifies whether the containers in the group will use the default security policy. Choosing "No" will force the containers in the group to use the custom security policy, defined below

Security Policy Settings: These settings create a default policy for the Container Group.

- **Token on Request:** this setting determines whether SOLA will accept requests without an attached security token.
  - **NO:** SOLA will allow requests without security tokens
  - **MainframeID:** SOLA will require a mainframe user id as a security token.
  - **LDAP ID:** SOLA will require an LDAP user id as a security token.
  - **SAML:** SOLA will require SAML credentials as a security token.
  - **Restrict by IP:** whether only certain IP addresses can submit requests

- **Token on Response:** with the current version of SOLA, the only option is NO.

- **Password on Request:** this setting determines whether SOLA accepts requests that have a token, but no password
  - **Optional:** a password is not required (SOLA will accept requests without a password).
  - **Mandatory:** a password is required.

- **Password on Response:** with the current version of SOLA, the only option is NO.

- **Encrypt on Request:** this setting determines whether SOLA accepts requests that are not encrypted.
  - **Optional:** encryption is not required (SOLA will accept requests without encryption).
  - **Mandatory:** encryption is required.

- **Encrypt on Response:** this setting determines whether SOLA will encrypt responses
  - **Optional:** SOLA will not encrypt responses
  - **Mandatory:** SOLA will encrypt responses

- **Signature on Request:** this setting determines whether SOLA accepts requests without an attached signature.
  - **Optional:** attached signatures are not required (SOLA will accept requests without attached signatures).
  - **Mandatory:** the body of the SOAP request must be signed.

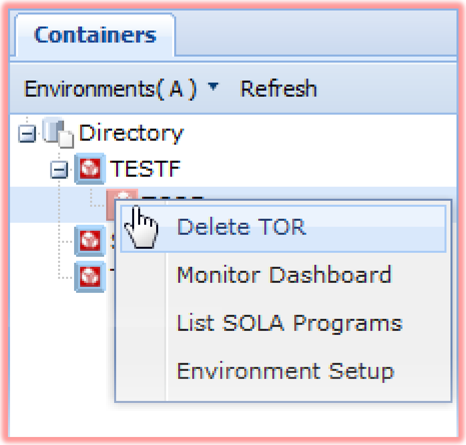
- **Signature on Response:** this setting determines whether SOLA will attach a signature to responses.
  - **Optional:** SOLA will not attach a signature to responses.
  - **Mandatory:** SOLA will attach a signature to responses.

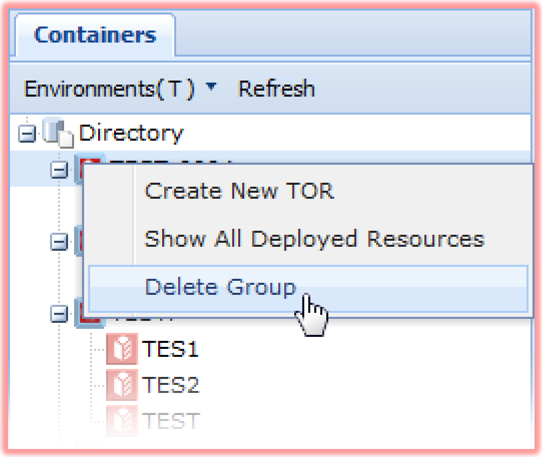
- **Timestamp on Request:** this setting determines whether SOLA accepts requests without an attached timestamp. The timestamp contains the policy’s expiration date and time.
  - **None:** attached timestamps are not required (SOLA will accept requests without attached timestamps).
  - **Mandatory:** attached timestamps are required.

- **Timestamp on Response:** with the current version of SOLA, the only option is NO.

When you are finished configuring the group, click **CREATE**. You can reset all the settings to their defaults at any time clicking the **RESET** button.

Deleting Containers and Container Groups



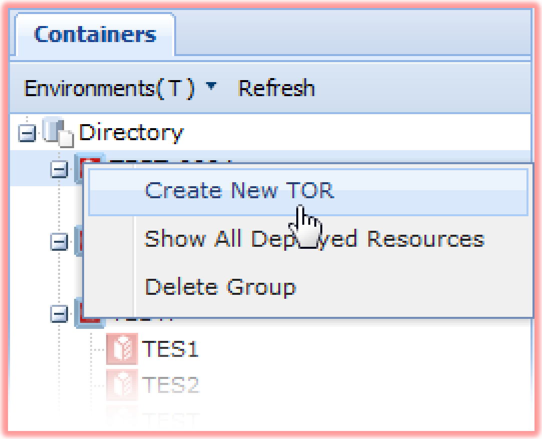


To delete a container, right click on the container and select **Delete TOR** from the menu.

To delete a group, right click the group and select **Delete Group** from the menu.

Deleting a group will delete all containers in that group and all group settings (custom security policy, etc.).

Creating Containers

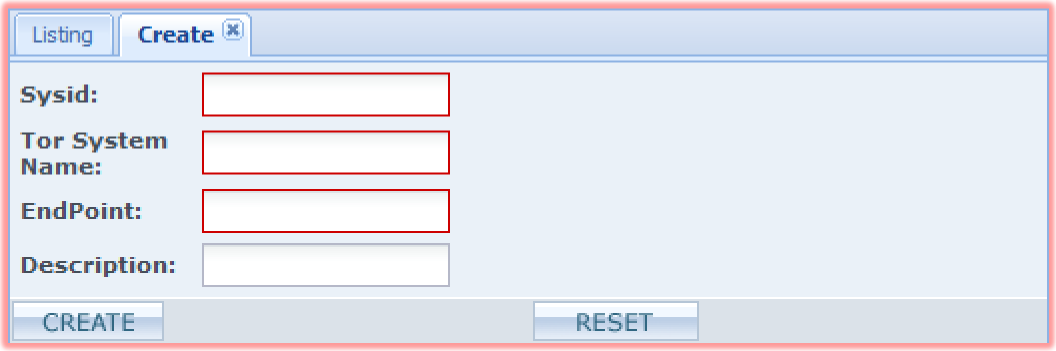


Once you have created and configured one or more container groups, you can create SOLA containers within those groups.

To create a SOLA container, right click on the group icon and select **Create New TOR** from the menu.

This will display the Create tab in the workspace, allowing you to create a new SOLA container.

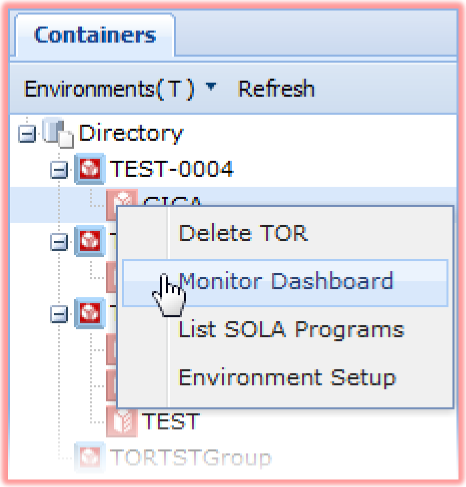
Fill out the required information about the container.



- **Sysid:** The 4 character SYSID of the region
- **Tor System Name:** The 8 character Applid of the region
- **EndPoint:** the region's IP address and port number that Container is listening to
- **Description**: a brief description of the region (optional).

When you have filled out all required fields, click **CREATE** to create the new container.

Monitoring SOLA Containers

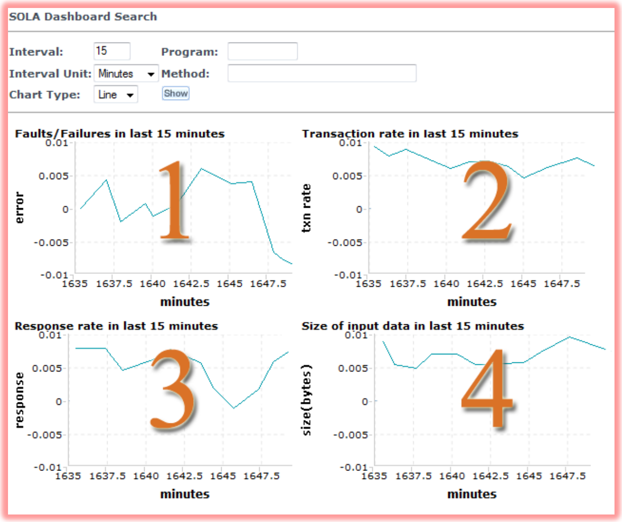


Resource Manager has an active dashboard that provides container performance metrics in real time. The metrics frequency is controlled by the dashboard, not the SOLA group settings for metrics collection.

To access the dashboard for a container, right click on the container and select **Monitor Dashboard** from the menu.

The Dashboard tab will be displayed in the workspace.

The dashboard is divided into four panels, each of which provides specific information (default settings shown).


SOLA Dashboard Search

Interval: 15 Program:

Interval Unit: Minutes Method:

Chart Type: Line

Show
Faults/Failures in last 15 minutes
error
0.01
0.005
0
-0.005
-0.01
1635 1637.5 1640 1642.5 1645 1647.5
minutes
1
Transaction rate in last 15 minutes
txn rate
2
Response rate in last 15 minutes
response
3
Size of input data in last 15 minutes
size(bytes)
4

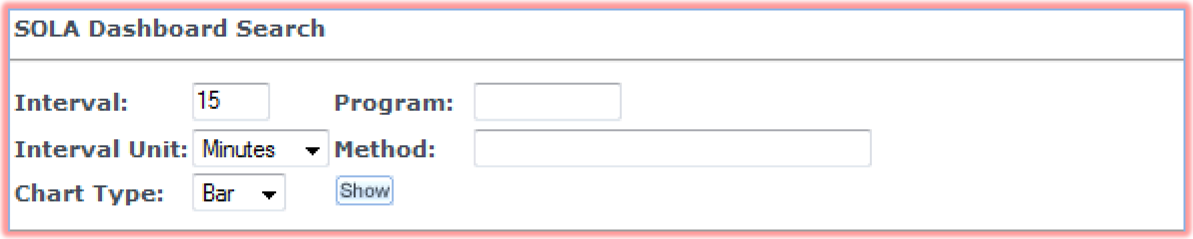
Panel 1: Faults/failure rate in the last *n minutes/seconds*.

Panel 2: Transaction rate in the last *n minutes/seconds*.

Panel 3: Response rate in the last *n minutes/seconds*.

Panel 4: Input data size over the last *n minutes/seconds*.

You can configure the dashboards using the top panel.


SOLA Dashboard Search

Interval: 15 Program:

Interval Unit: Minutes Method:

Chart Type: Bar

- **Interval:** the interval to chart, in minutes or seconds, depending on the Interval Unit menu.
- **Interval Unit:** determines whether the data collection interval is measured in minutes or seconds.
- **Chart Type:** Determines how the data is presented. Options are line or bar.
- **Program:** narrows down the data collection to a single program running in the container.
- **Method:** narrows down the data collection to a single method (operation) running in the container.

Establishing an LDAP server for authorizing LDAP credentials

LDAP

The Lightweight Directory Access Protocol, or LDAP, is an application protocol for querying and modifying directory services running over TCP/IP. SOLA supports the use of an LDAP server for authorization of LDAP credentials. The connection to the LDAP server can be accomplished using TLS (Transport Layer Security) for secure connections.

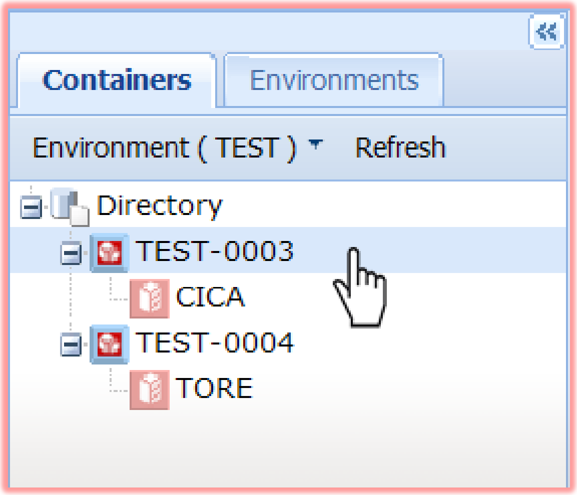
The definition of an LDAP connection is done at the SOLA Container Group, allowing multiple LDAP servers to be used (for example for Development, QA and Production).

Definition is a two step process.

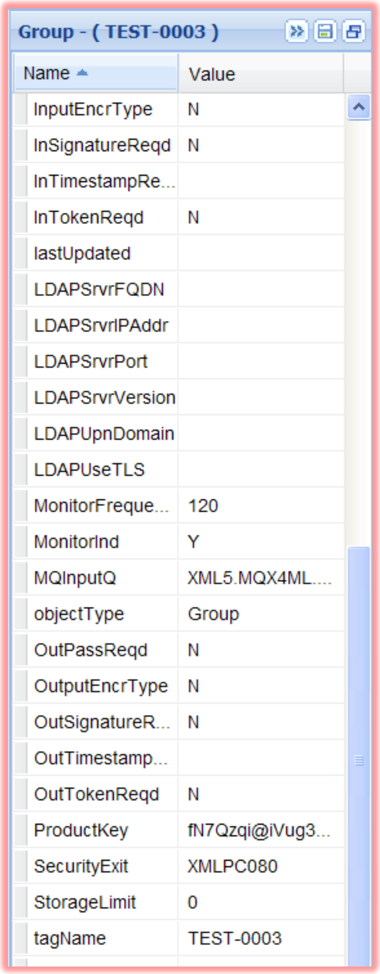
1. Define the Container Group, see Creating Container Groups on page 129
2. Update the properties for the Container Group to define the LDAP server.

Update a Container Group's properties

Using the SOLA Resource Manager, select a Container Group from the Containers List.



The Properties for the Container Group will be displayed in the Properties window (right pane).



Group - (TEST-0003)

Name	Value
InputEncrType	N
InSignatureReqd	N
InTimestampRe...	
InTokenReqd	N
lastUpdated	
LDAPSrvrFQDN	
LDAPSrvrIPAddr	
LDAPSrvrPort	
LDAPSrvrVersion	
LDAPUpnDomain	
LDAPUseTLS	
MonitorFreque...	120
MonitorInd	Y
MQInputQ	XML5.MQX4ML....
objectType	Group
OutPassReqd	N
OutputEncrType	N
OutSignatureR...	N
OutTimestamp...	
OutTokenReqd	N
ProductKey	fN7Qzqi@iVug3...
SecurityExit	XMLPC080
StorageLimit	0
tagName	TEST-0003

LDAPSrvrFQDN: LDAP Server hostname

LDAPSrvrIPAddr: Optional LDAP Server IP Address as an alternate address to FQDN

LDAPSrvrPort: LDAP Server port to connect to for validations. Port 389 is the well known port for LDAP non-SSL connection and Port 636 is the well known port for LDAP SSL connections.

LDAPSrvrVersion: Supported LDAP Versions are 2 and 3.

LDAPUpnDomain: Optional Default UPN(UserPrincipalName) Domain.
All LDAP Userids processed by SOLA confirm to the formats "\Username" or "UpnDomain\Username".

Backslash (****) in the <wsse:Username> element of the soap request is used to differentiate LDAP userids from SAF userids. If UpnDomain is not specified in the LDAP Userid then default UPN domain is prefixed for authenticating.

LDAPUseTLS: Specify 'Y' or 'N'.
If the LDAP port specified is an SSL port then specify 'Y' to indicate that AT-TLS on the mainframe is to be used for SSL communication.
Refer to z/OS Communication Server: IP Configuration Guide on how to configure Application Transparent Transport Layer Security (AT-TLS).

Monitoring and Error Logging Database Tables and Services

Monitoring

The SOLA monitoring feature, if enabled, records runtime metrics for every Web Service request that runs through SOLA. For information on how to turn monitoring on or off, refer to page 14.

For details on the metrics that are captured in the SOLA monitor see the SOLA Monitor Item Screen section of the SOLA User Guide.

For performance reasons, SOLA runtime metrics are written to an in-core Table.   A background process is responsible for offloading the metrics data from the in-core table to a corresponding DB2 table.

Since data is written in core at execution time, Web Service requests do not pay any measurable response time penalty for monitoring. For this reason it is recommended that monitoring should always be turned on.

Monitor DB2 Table

As mentioned, a SOLA background task unloads monitor data to DB2. This data is stored in a DB2 table called TBXMLMON. Following is the definition, and description, of this table.

Table: TBXMLMON - (SOLA monitor table)

Column Name	Type
XCTOR_SYS_ID	CHAR(4)
XCTOR_TRAN_ID	CHAR(4)
XCTOR_TSK_NO	DECIMAL(7,0)
XCAOR_SYS_ID	CHAR(4)
XCAOR_TSK_NO	DECIMAL(7,0)

XCDT_MTHD_NM	CHAR(35)
XCDT_PROG_NM	CHAR(8)
XCDT_PGM_TY_CD	CHAR(35)
XCDT_REQR_IP_AD	CHAR(15)
XCDT_TSK_STRT_DT	DATE(4)
XCDT_TSK_STRT_TM	TIME
XCDT_TSK_AOR_TM	INTEGER
XCDT_TSK_ELPS_TM	INTEGER
XCDT_TSK_CMP_CD	SMALLINT
XCDT_TSK_ABND_CD	CHAR(4)
XCDT_REQ_SZ	INTEGER
XCDT_RESP_SZ	INTEGER
USERTOKEN	CHAR(128)

The following are descriptions of each column:

- **XCTOR_SYS_ID:** This field holds the Terminal Owning Container System Id. This identifies the SOLA container that captured the initial SOAP request (the container where SOLA product runs).
- **XCTOR_TRAN_ID:** The transaction that handled the request.
- **XCTOR_TSK_NO:** The task number of the transaction that handled the request.
- **XCAOR_SYS_ID:** This field holds the Application Owning Container (AOR) System Id. This identifies the region where the legacy program that hosts web service resides.
- **XCAOR_TSK_NO:** The task number of the transaction that handled the request.
- **XCDT_MTHD_NM:** The method name associated with the request.
- **XCDT_PROG_NM:** The legacy program that is executed on behalf of the SOAP request.
- **XCDT_PGM_TY_CD:** The type of legacy program (e.g. 3270, COMMAREA, etc.).
- **XCDT_REQR_IP_AD:** The IP address of the requestor.
- **XCDT_TSK_STRT_DT:** The date that the request was received.
- **XCDT_TSK_STRT_TM:** The time that the request was received.
- **XCDT_TSK_AOR_TM:** The elapsed time, in milliseconds, consumed by the legacy application.
- **XCDT_TSK_ELPS_TM:** The total elapsed time, on the mainframe, for both the legacy application and SOLA.
- **XCDT_TSK_CMP_CD:** The request completion code.
- **XCDT_TSK_ABND_CD:** The ABEND code, if the process abended.
- **XCDT_REQ_SZ:** The SOAP request size.
- **XCDT_RESP_SZ:** The SOAP response size.

- **USERTOKEN:** The security token, if any, attached to the request (e.g. UserId or SAML token).

Services Provided on the Monitor Table Data

It is possible to use SOLA to create your own Web Service(s) to retrieve metrics data from the TBXMLMON table. However, there are two services included natively in SOLA for retrieving this data. One service, called getMonitorList, provides for the retrieval of monitor data based on a variety of input parameters. A second service called getMonitorItem will retrieve more detailed information for a particular SOLA transaction.

The WSDL for each of these operations can be found on the SOLA installation disk.

getMonitorList

This service produces a list of up to 200 monitor records. Each monitor record contains information regarding a single Web Service request processed by the SOLA runtime component. A variety of input parameters allow you to narrow the scope of the search. Following is a description of both the input and output parameters for this service.

Inputs:

- **RequestType:** Always '**Mon**'. This is the only required field.
- **IMonStartDateFrom:** Monitor data from this date (**yyyy-mm-dd**) will be retrieved.
- **IMonStartDateTo:** Monitor data to this date (**yyyy-mm-dd**) will be retrieved.
- **IMonStartTimeFrom:** Monitor Data from this time (**hh.mm.ss**) will be retrieved.
- **IMonStartTimeTo:** Monitor Data to this time (**hh.mm.ss**) will be retrieved.
- **IMonProgramType:** Retrieve Monitor statistics for this program type. **If supplied, program type must be specified in upper case.** Valid types are:

  - CA – Commarea programs
  - BM – BMS 3270 programs
  - CU – Custom DOM API Programs
  - SQ – Adhoc SQL Plug-in requests
  - SP – DB2 Stored Procedures
  - VS – VSAM plug-in requests

- **IMonProgramName:** If supplied, monitor data relating only to this legacy program will be returned. **Program, if supplied, name must be specified in upper case.**

- **IMonMethodName:** If supplied, monitor data relating only to this method will be returned. **Method name, if supplied, is case sensitive.**

- **IMonRequestIPAddr:** If supplied, monitor data relating only to this requestor's IP address will be returned.

- **IMonTorSysid:** If supplied, monitor data relating only to this Container will be returned. **Must be upper case.**

- **IMonTorTranid:** If supplied, monitor data relating only to this Container Transaction ID will be returned. **Must be upper case.**

- **IMonTorTaskNo:** If supplied, monitor data relating only to this Container Task number will be returned.

- **IMonAorSysid:** If supplied, monitor data relating only to legacy programs running in this AOR will be returned. **Must be upper case.**

- **IMonAorTaskNo:** If supplied, monitor data relating only to this AOR Task number will be returned.

TECH TIP

All of these fields are optional (except for RequestType) but it is strongly recommended that you provide the start and end date and time for performance reasons. Also, the start and end date and time should represent a relatively small interval (one or two hours for instance). Fields are case sensitive (see each field description for details).

Outputs:

- **ReturnCode:** Execution status. This will be zero for successful execution.

- **ReturnMsg:** Return Message. This will be blank when ReturnCode is zero. If ReturnCode is not zero is will contain an error message.

- **FetchCounter:** Total number of monitor records retrieved.

- **ReturnCICSCode:** In the case of a CICS failure, this will be the EIBRESP code.

- **ReturnDB2Code:** In the case of a DB2 failure, this will be the SQLCODE.

- **OMonResultTable:** There will be one of these tags for each row returned. The data elements in each row will be attributes of this tag.

- **OMonStartDate:** The date (yyyy-mm-dd) of the transaction represented by this monitor row.

- **OMonStartTime:** The execution time (hh.mm.ss) of the transaction represented by this monitor row.

- **OMonProgramType:** The type of legacy program that hosted the web service request.

- **OMonProgramName:** The name of the legacy program that hosted the web service request.

- **OMonMethodName:** Service's Method/Operation Name.

- **OMonRequestIPAddr:** The IP address of the client that initiated the Web Service request.

- **OMonTorSysid:** The Container's SYSID of the container that received the request.

- **OMonTorTranid:** The transaction ID of the Container's transaction that handled the request.

- **OMonTorTaskNo:** The task number of the Container transaction that handled the request.

- **OMonAorSysid:** The AOR's SYSID of the container that received the request.

- **OMonAorTaskNo:** The task number of the AOR transaction that handled the request.

- **OMonTaskAorTime:** The elapsed time, in milliseconds, consumed by the legacy program running in the AOR container.

- **OMonTaskElapsedTime:** The total task elapsed time, in milliseconds, for both AOR and TOR containers. This includes the time consumed by the SOLA runtime components.

- **OMonTaskCompCode:** The Web Services completion code. '200' is normal completion.

- **OMonTaskAbendCode:** If the legacy program, or a SOLA component, abends during processing, then this will be the program abend code (ASRA for instance).

- **OMonRequestSize:** The size of the inbound SOAP request.

- **OMonResponseSize:** The size of the outbound SOAP response.

getMonitorItem

This service will provide monitor details for a single Web Service request that was processed by the SOLA runtime component. The resulting record will contain information regarding the Web Service request. The following is a description of both the input and output parameters for this service.

Inputs:

- **RequestType:** Always 'Mon'. This is the only required field.

- **IMonStartDate:** The date (yyyy-mm-dd) that the requested Web Service was run.

- **IMonStartTime:** The time (hh.mm.ss) that the requested Web Service was run.

- **IMonProgramName:** The legacy program name (**must be upper case**) that hosted the Web Service.

- **IMonMethodName:** The Web Service's Method Name (**case sensitive**).

- **IMonTorTaskNo:** The task number for that task that handled the Web Service request.

SOLA™ TECH TIP

All of these fields are required and case sensitive (see field descriptions for case requirements). Proper values can be determined by using the 'getMonitorList' service.

Outputs:

Outputs are the same as getMonitorList, except that only one row of data from the monitor table will be returned.

Error Logging

In addition to being captured in the SOLA monitor, failed transactions are recorded in the SOLA Error Log. The error log is a DB2 table called TBXMLLOG. Unlike monitor data, error log data is written directly to DB2. Since errors represent exception conditions, it wasn't deemed necessary to write them to an in-core table for performance reasons. There is no administrative option to turn logging off. It is always active.

The Error Log table is also used whenever auditing is turned on for a particular web service operation. Auditing will cause SOAP requests to be written to the Error Log table in the event that inbound auditing is turned on for the operation. If outbound auditing is turned on for the operation, then SOAP responses are written to the Error Log table.

In the case of inbound auditing, the following message will appear in the ERROR_MSG column of the table:

Audit Record (SOAP Request) (TQCL/XMLPC000)

For outbound auditing, the following message will appear in the ERROR_MSG column of the table:

Audit Record (SOAP Response) (TQCL/XMLPC000)

SOLA™ TECH TIP

If either inbound or outbound auditing is turned on for a particular web service operation, the appropriate SOAP messages are written to the Error Log table even if the web service operation completes successfully. This implies that auditing will have a slight performance impact on the web service and, therefore, should be left on only for short a duration.

The contents of the Error Log are described in the following section.

Error Log DB2 Table

Table: TBXMLLOG - (SOLA Error Log table)

Column Name	Type
LOG_TS	TIMESTMP
METHOD_NM	CHAR(35)
PROG_NM	CHAR(8)
PROG_TY	CHAR(35)
ERROR_CD	SMALLINT
XCTOR_TSK_NO	DECIMAL(7,0)
ERROR_MSG	VARCHAR(254)
SOAP_REQ	LONGVAR(3696)

The following are descriptions of each column:

- **LOG_TS:** The DB2 timestamp when the error was recorded.
- **METHOD_NM:** Method Name for the Web Service that caused the exception.
- **PROG_NM:** Legacy program targeted by the Web Service.
- **PROG_TY:** Retrieve Monitor statistics for this program type. Valid types are:
  - CA – Commarea programs
  - BM – BMS 3270 programs
  - CU – Custom DOM API Programs
  - SQ – Adhoc SQL Plug-in requests
  - SP – DB2 Stored Procedures
  - VS – VSAM plug-in requests
- **ERROR_CD:** Not currently used. Always zero.
- **XCTOR_TSK_NO:** The task number of the transaction that handled the request.
- **ERROR_MSG:** Description of the error.

- **SOAP_REQ:** The inbound SOAP request that caused the error. If outbound auditing was turned on for the Web Service then there would be two rows in this table. The first would contain the inbound SOAP request in this column and the second would contain the outbound SOAP response.

Services provided on the Error Log Table Data

It is possible to use SOLA to create your own Web Service(s) to retrieve error information from the TBXMLLOG table. However, there are two services included natively in the SOLA product for retrieving this data. One service, called getErrorList, provides for retrieval of Error Log data based on a variety of input parameters. A second service, getErrorItem, retrieves more detailed information for a particular error.

The WSDL for each of these operations can be found in the SOLA installation downloaded zip file.

getErrorList

This service will produce a list of up to 275 error records. Each record contains information regarding a single Web Service request processed by the SOLA runtime component. A variety of input parameters allow you to narrow the scope of the search. The following is a description of both the input and output parameters for this service.

Inputs:

- **RequestType:** Always **Log**. This is the only required field.

- **ILogTimeStampFrom:** Error Log data from this date/time (**yyyy-mm-dd-hh.mm.ss.ssssss**) will be retrieved. This field is used in conjunction with ILogTimeStampTo.

- **ILogTimeStampTo:** Error Log data to this date/time (**yyyy-mm-dd-hh.mm.ss.ssssss**) will be retrieved. This field is used in conjunction with ILogTimeStampFrom.

- **ILogProgramType:** Retrieve Error Log statistics for this program type. **If supplied, program type must be specified in upper case.** Valid types are:

  - CA – Commarea programs

  - BM – BMS 3270 programs
  - CU – Custom DOM API Programs
  - SQ – Adhoc SQL Plug-in requests
  - SP – DB2 Stored Procedures
  - VS – VSAM plug-in requests

- **ILogProgramName:** If supplied, Error Log data relating only to this legacy program will be returned. **Program, if supplied, name must be specified in upper case.**

- **ILogMethodName:** If supplied, Error Log data relating only to this method will be returned. **Method name, if supplied, is case sensitive.**

TECH TIP

All of these fields are optional (except for RequestType), but it is strongly recommended that you provide the start and end date time stamp for performance reasons. Also that start and end date and time should represent a relatively small interval (one or two hours for instance). Fields are case sensitive (see each field description for details).

Outputs:

- **ReturnCode:** Execution status. This will be zero for successful execution.

- **ReturnMsg:** Return Message. This will be blank when ReturnCode is zero. If ReturnCode is not zero it will contain an error message.

- **FetchCounter:** Total number of monitor records retrieved.

- **ReturnCICSCode:** In the case of a CICS failure, this will be the EIBRESP code.

- **ReturnDB2Code:** In the case of a DB2 failure, this will be the SQLCODE.

- **InternalMessage:** Additional error information.

- **OLogResultTable:** There will be one of these tags for each row returned. The data elements in each row will be attributes of this tag.

- **OLogTimeStamp:** The date/time stamp (yyyy-mm-dd-hh.mm.ss.ssssss) of the transaction represented by this error instance.

- **OLogProgramType:** The type of legacy program that hosted the web service in error.
- **OLogProgramName:** The name of the legacy program that hosted the web service in error.
- **OLogMethodName:** The service's Method/Operation Name.
- **OLogErrorCode:** This will always be zero.
- **OLogTorTaskNo:** The task number of the transaction that handled the request.

getErrorItem

This service will provide Error Log details on a single Web Service request that was processed by the SOLA runtime component. The resulting record will contain information regarding the Web Service request. The following is a description of both the input and output parameters for this service.

Inputs:

- **RequestType:** Always **Log**. This is the only required field.
- **ILogTimeStamp:** The DB2 Timestamp (yyyy-mm-dd-hh.mm.ss.ssssss) when the Web Service transaction was executed. The proper value for this field can be obtained from running the getErrorList service.

SOLA™ TECH TIP

Both of these fields are required. Proper values for ILogTimeStamp can be determined by using the getErrorList service.

Outputs:

- **ReturnCode:** Execution status. This will be zero for successful execution.
- **ReturnMsg:** Return Message. This will be blank when ReturnCode is zero. If ReturnCode is not zero it will contain an error message.
- **FetchCounter:** Total number of monitor records retrieved.

- **ReturnCICSCode:** In the case of a CICS failure, this will be the EIBRESP code.
- **ReturnDB2Code:** In the case of a DB2 failure, this will be the SQLCODE.
- **InternalMessage:** Additional error information.
- **OLogTimeStamp:** The date/time stamp (yyyy-mm-dd-hh.mm.ss.ssssss) of the transaction represented by this error instance.
- **OLogProgramType:** The type of legacy program that hosted the web service in error.
- **OLogProgramName:** The name of the legacy program that hosted the web service in error.
- **OLogMethodName:** The service's Method/Operation Name.
- **OLogErrorCode:** This will always be zero.
- **OLogTorTaskNo:** The task number of the transaction that handled the request.
- **OLogErrorMsg:** Description of the error.
- **OLogSoapRequest:** The SOAP request that caused the error.

Appendix A: SOLA DB2 Database

Database

The SOLA Directory consists of 28 tables created in a single database DBXML003. The creator of the tables is specified in the WRKSHEET as <*qualifier*>.

Database DDL

```
-----------------------------------------------------------------------
-- Stogroup=SOLASTG
-----------------------------------------------------------------------
```

```
--
  CREATE STOGROUP SOLASTG
    VOLUMES(SOLA01)
    VCAT SYSC ;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
------------------------------------------------------------------------
--
  CREATE DATABASE DBXML003
    BUFFERPOOL BP0
    INDEXBP    BP0
    CCSID      EBCDIC
    STOGROUP   SOLASTG;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLACC
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLACC
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLACC In DBXML003.TSXMLACC
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLACC
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPID              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      RESOURCEID           CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      RESOURCETYPE         CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      OPERATIONTYPE        CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLACC
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLACC On SOLA600P.TBXMLACC
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLACC
    ON SOLA600P.TBXMLACC
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLACC On SOLA600P.TBXMLACC
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLACC
    ON SOLA600P.TBXMLACC
     (GROUPID               ASC,
      RESOURCEID            ASC,
      RESOURCETYPE          ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
```

```
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLALT
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLALT
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLALT In DBXML003.TSXMLALT
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLALT
     (ID                     TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ALERTCODE              INTEGER NOT NULL WITH DEFAULT,
      METRICTYPE             INTEGER NOT NULL WITH DEFAULT,
      OPERATOR               CHAR(2) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      "VALUE"                INTEGER NOT NULL WITH DEFAULT,
      EFFECTIVE              TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES                TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL                 VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLALT
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLALT On SOLA600P.TBXMLALT
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLALT
    ON SOLA600P.TBXMLALT
     (ID                      ASC,
      EXPIRES                 ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLALT On SOLA600P.TBXMLALT
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLALT
    ON SOLA600P.TBXMLALT
     (ALERTCODE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLASN
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLASN
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
```

```
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLASN In DBXML003.TSXMLASN
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLASN
     (ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ID1                  TIMESTAMP NOT NULL WITH DEFAULT,
      TYPE1                CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ID2                  TIMESTAMP NOT NULL WITH DEFAULT,
      TYPE2                CHAR(26) FOR SBCS DATA NOT NULL,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLASN
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLASN On SOLA600P.TBXMLASN
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLASN
    ON SOLA600P.TBXMLASN
     (ENVIRONID             ASC,
      ID1                   ASC,
      TYPE1                 ASC,
      ID2                   ASC,
      TYPE2                 ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLCER
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLCER
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLCER In DBXML003.TSXMLCER
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLCER
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ISSUER               CHAR(128) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      CERTTYPE             CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      SUBJECT              CHAR(128) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      CERTSN               CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      CERTSEQ              SMALLINT NOT NULL WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      PUBLIC_KEY_EXP       CHAR(3) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PUBLIC_KEY_MOD       VARCHAR(256) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      DETAIL               VARCHAR(3300) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLCER
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLCER On SOLA600P.TBXMLCER
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLCER
    ON SOLA600P.TBXMLCER
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLCER On SOLA600P.TBXMLCER
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLCER
    ON SOLA600P.TBXMLCER
     (PUBLIC_KEY_MOD        ASC,
      CERTSEQ               ASC,
      CERTTYPE              ASC,
      EXPIRES               ASC,
      EFFECTIVE             ASC)
    NOT PADDED
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLCOM
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLCOM
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLCOM In DBXML003.TSXMLCOM
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLCOM
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      PROGRAMID            TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            TIMESTAMP NOT NULL WITH DEFAULT,
      COLUMNNM             CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ROWNUM               SMALLINT NOT NULL WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLCOM
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLCOM On SOLA600P.TBXMLCOM
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLCOM
    ON SOLA600P.TBXMLCOM
     (ID                    ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLCOM On SOLA600P.TBXMLCOM
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLCOM
    ON SOLA600P.TBXMLCOM
     (COLUMNNM              ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X3XMLCOM On SOLA600P.TBXMLCOM
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X3XMLCOM
    ON SOLA600P.TBXMLCOM
     (PROGRAMID             ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLENV
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLENV
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLENV In DBXML003.TSXMLENV
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLENV
     (ID                     TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONMENT            CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ENVIRONSEQ             SMALLINT NOT NULL WITH DEFAULT,
      EFFECTIVE              TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES                TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL                 VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLENV
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLENV On SOLA600P.TBXMLENV
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLENV
    ON SOLA600P.TBXMLENV
     (ID                      ASC,
      EXPIRES                 ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLENV On SOLA600P.TBXMLENV
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLENV
    ON SOLA600P.TBXMLENV
     (ENVIRONSEQ              ASC,
      ENVIRONMENT             ASC,
      EFFECTIVE               ASC,
      EXPIRES                 ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X3XMLENV On SOLA600P.TBXMLENV
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X3XMLENV
    ON SOLA600P.TBXMLENV
     (ENVIRONMENT           ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLEXT
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLEXT
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
```

```
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLEXT In DBXML003.TSXMLEXT
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLEXT
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPID              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EXITNAME             CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EXITTYPE             CHAR(3) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLEXT
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLEXT On SOLA600P.TBXMLEXT
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLEXT
    ON SOLA600P.TBXMLEXT
     (ID                    ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 15 PCTFREE 5
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLEXT On SOLA600P.TBXMLEXT
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLEXT
    ON SOLA600P.TBXMLEXT
     (EXITNAME              ASC,
      EXITTYPE              ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 15 PCTFREE 10
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLGRP
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLGRP
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLGRP In DBXML003.TSXMLGRP
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLGRP
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPNM              CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPTYPE            CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLGRP
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLGRP On SOLA600P.TBXMLGRP
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLGRP
    ON SOLA600P.TBXMLGRP
     (ID                      ASC,
      EXPIRES                 ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLGRP On SOLA600P.TBXMLGRP
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLGRP
    ON SOLA600P.TBXMLGRP
     (GROUPNM                 ASC,
      GROUPTYPE               ASC,
      ENVIRONID               ASC,
      EFFECTIVE               ASC,
      EXPIRES                 ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLIPA
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLIPA
```

```
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLIPA In DBXML003.TSXMLIPA
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLIPA
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPID              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      IP_AD                CHAR(15) FOR SBCS DATA NOT NULL,
      IP_NODE_1            CHAR(3) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      IP_NODE_2            CHAR(3) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      IP_NODE_3            CHAR(3) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      IP_NODE_4            CHAR(3) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      USAGE_STATS          INTEGER NOT NULL WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLIPA
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLIPA On SOLA600P.TBXMLIPA
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLIPA
    ON SOLA600P.TBXMLIPA
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLIPA On SOLA600P.TBXMLIPA
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLIPA
    ON SOLA600P.TBXMLIPA
     (IP_AD                 ASC,
      EXPIRES               ASC,
      EFFECTIVE             ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLLOG
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLLOG
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLLOG In DBXML003.TSXMLLOG
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLLOG
     (LOG_TS               TIMESTAMP NOT NULL WITH DEFAULT,
      METHOD_NM            CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PROG_NM              CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PROG_TY              CHAR(35) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ERROR_CD             SMALLINT NOT NULL WITH DEFAULT,
      XCTOR_TSK_NO         DECIMAL(7, 0) NOT NULL WITH DEFAULT,
      LOG_SEQ              SMALLINT NOT NULL WITH DEFAULT,
      ERROR_MSG            VARCHAR(254) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      SOAP_REQ             VARCHAR(3600) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLLOG
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLLOG On SOLA600P.TBXMLLOG
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLLOG
    ON SOLA600P.TBXMLLOG
     (LOG_TS                ASC,
      PROG_NM               ASC,
      METHOD_NM             ASC,
      PROG_TY               ASC,
      LOG_SEQ               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLMAP
```

```
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLMAP
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 48 SECQTY 48
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD NO
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLMAP In DBXML003.TSXMLMAP
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLMAP
     (PROG                  CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ENVIRONID             TIMESTAMP NOT NULL WITH DEFAULT,
      METHOD                CHAR(40) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ROWNUM                INTEGER NOT NULL WITH DEFAULT,
      MAPNAME               CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MAPSET                CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      TRANS                 CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      TOTALINPUT            INTEGER NOT NULL WITH DEFAULT,
      REPEATCOUNT           INTEGER NOT NULL WITH DEFAULT,
      RECEIVETYPE           CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MAPKEY                CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MAP_WIDTH             SMALLINT NOT NULL WITH DEFAULT,
      MAP_HEIGHT            SMALLINT NOT NULL WITH DEFAULT,
      TOTAL_FIELDS          SMALLINT NOT NULL WITH DEFAULT,
      EXTENDED_ATTR         CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      LAST_MAP              CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      NO_EXTENDED_ATTR      SMALLINT NOT NULL WITH DEFAULT,
      MAP_DDKEY             CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      DRILLDOWNTYPE         CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT)
    IN DBXML003.TSXMLMAP
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLMAP On SOLA600P.TBXMLMAP
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLMAP
    ON SOLA600P.TBXMLMAP
     (PROG                  ASC,
      ENVIRONID             ASC,
      METHOD                ASC,
      ROWNUM                ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 5712 SECQTY 48
    ERASE  NO
    FREEPAGE 3 PCTFREE 25
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE YES
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLMON
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLMON
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 1440 SECQTY 7200
    ERASE  NO
    FREEPAGE 15 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
```

```
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLMON In DBXML003.TSXMLMON
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLMON
     (XCTOR_SYS_ID         CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCTOR_TRAN_ID        CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCTOR_TSK_NO         DECIMAL(7, 0) NOT NULL WITH DEFAULT,
      XCAOR_SYS_ID         CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCAOR_TSK_NO         DECIMAL(7, 0) NOT NULL WITH DEFAULT,
      XCDT_MTHD_NM         CHAR(35) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_PROG_NM         CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_PGM_TY_CD       CHAR(35) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_REQR_IP_AD      CHAR(15) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_TSK_STRT_DT     DATE NOT NULL WITH DEFAULT,
      XCDT_TSK_STRT_TM     TIME NOT NULL WITH DEFAULT,
      XCDT_TSK_AOR_TM      INTEGER NOT NULL WITH DEFAULT,
      XCDT_TSK_ELPS_TM     INTEGER NOT NULL WITH DEFAULT,
      XCDT_TSK_CMP_CD      SMALLINT NOT NULL WITH DEFAULT,
      XCDT_TSK_ABND_CD     CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_REQ_SZ          INTEGER NOT NULL WITH DEFAULT,
      XCDT_RESP_SZ         INTEGER NOT NULL WITH DEFAULT,
      USERTOKEN            CHAR(128) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLMON
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLMON On SOLA600P.TBXMLMON
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X1XMLMON
    ON SOLA600P.TBXMLMON
     (XCDT_TSK_STRT_DT      ASC,
      XCDT_TSK_STRT_TM      ASC,
      XCDT_PROG_NM          ASC,
      XCDT_MTHD_NM          ASC,
      XCTOR_TSK_NO          ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 15 PCTFREE 5
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLMSK
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLMSK
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLMSK In DBXML003.TSXMLMSK
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLMSK
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPID              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MASKNM               CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MASKTYPE             CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      USAGESTATS           INTEGER NOT NULL WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLMSK
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
```

```
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLMSK On SOLA600P.TBXMLMSK
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLMSK
    ON SOLA600P.TBXMLMSK
     (ID                    ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLMSK On SOLA600P.TBXMLMSK
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLMSK
    ON SOLA600P.TBXMLMSK
     (GROUPID               ASC,
      MASKNM                ASC,
      MASKTYPE              ASC,
      EXPIRES               ASC,
      EFFECTIVE             ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLMTD
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLMTD
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLMTD In DBXML003.TSXMLMTD
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLMTD
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            TIMESTAMP NOT NULL WITH DEFAULT,
      PROGRAMID            TIMESTAMP NOT NULL WITH DEFAULT,
      POLICYID             CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      METHODNM             CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      TEMPLATENM           CHAR(8) FOR SBCS DATA WITH DEFAULT NULL,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLMTD
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLMTD On SOLA600P.TBXMLMTD
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLMTD
    ON SOLA600P.TBXMLMTD
     (ID                    ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLMTD On SOLA600P.TBXMLMTD
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLMTD
    ON SOLA600P.TBXMLMTD
     (METHODNM              ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X3XMLMTD On SOLA600P.TBXMLMTD
------------------------------------------------------------------------
--
  CREATE UNIQUE WHERE NOT NULL INDEX SOLA600P.X3XMLMTD
    ON SOLA600P.TBXMLMTD
     (TEMPLATENM            ASC,
      ENVIRONID             ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLMTL
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLMTL
```

```
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLMTL In DBXML003.TSXMLMTL
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLMTL
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      METHODID             TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            TIMESTAMP NOT NULL WITH DEFAULT,
      SCHEMANM             CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ROWNUM               SMALLINT NOT NULL WITH DEFAULT,
      IO                   CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PROGRAMID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      CTXSNSTIVEID         CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLMTL
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLMTL On SOLA600P.TBXMLMTL
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLMTL
    ON SOLA600P.TBXMLMTL
     (ID                    ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLMTL On SOLA600P.TBXMLMTL
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLMTL
    ON SOLA600P.TBXMLMTL
     (SCHEMANM              ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X3XMLMTL On SOLA600P.TBXMLMTL
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X3XMLMTL
    ON SOLA600P.TBXMLMTL
     (METHODID              ASC,
      IO                    ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
```

```
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X4XMLMTL On SOLA600P.TBXMLMTL
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X4XMLMTL
    ON SOLA600P.TBXMLMTL
     (PROGRAMID             ASC,
      CTXSNSTIVEID          ASC,
      EFFECTIVE             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLMTS
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLMTS
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 3600 SECQTY 1800
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD NO
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLMTS In DBXML003.TSXMLMTS
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLMTS
     (PROG                 CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ENVIRONID            TIMESTAMP NOT NULL WITH DEFAULT,
      METHOD                 CHAR(40) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MAPNAME                CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      MAPSET                 CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      CNAME                  CHAR(30) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ROWNUM                 INTEGER NOT NULL WITH DEFAULT,
      ROWPOSINMAP            SMALLINT NOT NULL WITH DEFAULT,
      IO                     CHAR(12) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ATTRDEF                CHAR(12) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ATTR2                  CHAR(2) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PROT_IND               SMALLINT NOT NULL WITH DEFAULT,
      HIDDEN_IND             SMALLINT NOT NULL WITH DEFAULT,
      MDT_IND                SMALLINT NOT NULL WITH DEFAULT,
      LEN                    SMALLINT NOT NULL WITH DEFAULT,
      "ROW"                  SMALLINT NOT NULL WITH DEFAULT,
      COLM                   SMALLINT NOT NULL WITH DEFAULT,
      VAL                    CHAR(100) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      RECV_TYP               CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      SPCL_IN                CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GP_IN                  CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      SPLIT_PARENT           SMALLINT NOT NULL WITH DEFAULT,
      MAP_INSTANCE           SMALLINT NOT NULL WITH DEFAULT,
      PARENT                 CHAR(30) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE              TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES                TIMESTAMP NOT NULL WITH DEFAULT)
    IN DBXML003.TSXMLMTS
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLMTS On SOLA600P.TBXMLMTS
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLMTS
    ON SOLA600P.TBXMLMTS
     (PROG                    ASC,
      ENVIRONID               ASC,
      METHOD                  ASC,
      ROWNUM                  ASC,
      EXPIRES                 ASC)
    USING STOGROUP SOLASTG
    PRIQTY 5712 SECQTY 48
    ERASE  NO
    FREEPAGE 3 PCTFREE 25
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE YES
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLOFT
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLOFT
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 16576 SECQTY 8288
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP32K1
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLOFT In DBXML003.TSXMLOFT
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLOFT
     (XCAOR_SYS_ID         CHAR(4) FOR SBCS DATA NOT NULL,
      XCAOR_TSK_NO         DECIMAL(7, 0) NOT NULL,
      XCDT_SQ_NO           SMALLINT NOT NULL,
      XCTOR_SYS_ID         CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCTOR_TSK_NO         DECIMAL(7, 0) NOT NULL WITH DEFAULT,
      CICSP_NM             CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_PGM_TY_CD       CHAR(35) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      XCDT_TSK_STRT_TS     TIMESTAMP NOT NULL WITH DEFAULT,
      XCDT_OVL_DATA        VARCHAR(32000) FOR SBCS DATA NOT NULL)
    IN DBXML003.TSXMLOFT
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLOFT On SOLA600P.TBXMLOFT
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X1XMLOFT
    ON SOLA600P.TBXMLOFT
     (XCAOR_TSK_NO          ASC,
      XCTOR_TSK_NO          ASC,
      XCDT_SQ_NO            ASC)
    USING STOGROUP SOLASTG
    PRIQTY 3600 SECQTY 1800
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE YES
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLPGM
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLPGM
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLPGM In DBXML003.TSXMLPGM
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLPGM
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      PROJECTID            TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            TIMESTAMP NOT NULL WITH DEFAULT,
      POLICYID                 CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PROGRAMNM                CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      CLASSNM                  CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE                TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES                  TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL                   VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLPGM
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLPGM On SOLA600P.TBXMLPGM
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLPGM
    ON SOLA600P.TBXMLPGM
     (ID                        ASC,
      ENVIRONID                 ASC,
      EXPIRES                   ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLPGM On SOLA600P.TBXMLPGM
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLPGM
    ON SOLA600P.TBXMLPGM
     (PROGRAMNM                 ASC,
      PROJECTID                 ASC,
      ENVIRONID                 ASC,
      EXPIRES                   ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLPOL
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLPOL
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLPOL In DBXML003.TSXMLPOL
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLPOL
     (ID                    TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID             CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      POLICYNM              CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      POLICYPOINT           CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      POLICYIOIND           CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      POLICYSEQ             SMALLINT NOT NULL WITH DEFAULT,
      EFFECTIVE             TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES               TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL                VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLPOL
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
```

```
-- Database=DBXML003
--    Index=SOLA600P.X1XMLPOL On SOLA600P.TBXMLPOL
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLPOL
    ON SOLA600P.TBXMLPOL
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLPOL On SOLA600P.TBXMLPOL
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLPOL
    ON SOLA600P.TBXMLPOL
     (POLICYNM              ASC,
      POLICYPOINT           ASC,
      POLICYIOIND           ASC,
      POLICYSEQ             ASC,
      EXPIRES               ASC,
      EFFECTIVE             ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLPRJ
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLPRJ
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLPRJ In DBXML003.TSXMLPRJ
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLPRJ
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      PROJECTNM            CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLPRJ
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLPRJ On SOLA600P.TBXMLPRJ
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLPRJ
    ON SOLA600P.TBXMLPRJ
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLPRJ On SOLA600P.TBXMLPRJ
```

```
------------------------------------------------------------------------
--
  CREATE INDEX SOLA600P.X2XMLPRJ
    ON SOLA600P.TBXMLPRJ
     (PROJECTNM              ASC,
      EFFECTIVE              ASC,
      EXPIRES                ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLSCH
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLSCH
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLSCH In DBXML003.TSXMLSCH
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLSCH
     (ID                     TIMESTAMP NOT NULL WITH DEFAULT,
      CLASSTYPE              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PROPERTYNM             CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ROWNUM                 SMALLINT NOT NULL WITH DEFAULT,
      EFFECTIVE              TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES                TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLSCH
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLSCH On SOLA600P.TBXMLSCH
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLSCH
    ON SOLA600P.TBXMLSCH
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLSCH On SOLA600P.TBXMLSCH
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLSCH
    ON SOLA600P.TBXMLSCH
     (CLASSTYPE             ASC,
      PROPERTYNM            ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLSPT
------------------------------------------------------------------------
```

```
--
  CREATE TABLESPACE TSXMLSPT
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 48 SECQTY 48
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD NO
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLSPT In DBXML003.TSXMLSPT
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLSPT
     (PROG                  CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ENVIRONID             TIMESTAMP NOT NULL WITH DEFAULT,
      METHOD                CHAR(40) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      ROWNUM                INTEGER NOT NULL WITH DEFAULT,
      MAPNO                 SMALLINT NOT NULL WITH DEFAULT,
      PARENTNO              SMALLINT NOT NULL WITH DEFAULT,
      SPLITNAME             CHAR(30) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      OFFSET                INTEGER NOT NULL WITH DEFAULT,
      LENGTH                INTEGER NOT NULL WITH DEFAULT,
      EFFECTIVE             TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES               TIMESTAMP NOT NULL WITH DEFAULT)
    IN DBXML003.TSXMLSPT
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLSPT On SOLA600P.TBXMLSPT
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLSPT
    ON SOLA600P.TBXMLSPT
     (PROG                   ASC,
      ENVIRONID              ASC,
      METHOD                 ASC,
      ROWNUM                 ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 5712 SECQTY 48
    ERASE  NO
    FREEPAGE 3 PCTFREE 25
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE YES
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLTOR
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLTOR
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLTOR In DBXML003.TSXMLTOR
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLTOR
     (ID                    TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID             CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPID               CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      SYSID                 CHAR(4) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      TORNM                 CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE             TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES               TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL                VARCHAR(3800) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLTOR
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLTOR On SOLA600P.TBXMLTOR
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLTOR
    ON SOLA600P.TBXMLTOR
     (ID                    ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLTOR On SOLA600P.TBXMLTOR
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLTOR
    ON SOLA600P.TBXMLTOR
     (SYSID                 ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLUAC
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLUAC
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 7200 SECQTY 3600
    ERASE  NO
    FREEPAGE 4 PCTFREE 25
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 32
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLUAC In DBXML003.TSXMLUAC
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLUAC
     (UAC_MAJORAID         CHAR(4) FOR SBCS DATA NOT NULL,
      UAC_MINORAID         CHAR(3) FOR SBCS DATA NOT NULL,
      UAC_ROWNUMBR         INTEGER NOT NULL,
      UAC_USERNAME         CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      UAC_USERDATE         DATE NOT NULL WITH DEFAULT,
      UAC_USERTIME         TIME NOT NULL WITH DEFAULT,
      UAC_ENDPOINT         CHAR(54) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLUAC
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLUAC On SOLA600P.TBXMLUAC
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLUAC
    ON SOLA600P.TBXMLUAC
     (UAC_MAJORAID          ASC,
      UAC_MINORAID          ASC,
      UAC_ROWNUMBR          ASC)
    USING STOGROUP SOLASTG
    PRIQTY 7200 SECQTY 1440
    ERASE  NO
    FREEPAGE 4 PCTFREE 25
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLUAP
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLUAP
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 3600 SECQTY 1800
    ERASE  NO
    FREEPAGE 4 PCTFREE 25
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 32
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 1;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLUAP In DBXML003.TSXMLUAP
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLUAP
     (UAP_MAJORAID         CHAR(4) FOR SBCS DATA NOT NULL,
      UAP_MINORAID         CHAR(3) FOR SBCS DATA NOT NULL,
      UAP_SAVECNTR         INTEGER NOT NULL WITH DEFAULT,
      UAP_MAXLIMIT         INTEGER NOT NULL WITH DEFAULT)
    IN DBXML003.TSXMLUAP
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLUAP On SOLA600P.TBXMLUAP
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLUAP
    ON SOLA600P.TBXMLUAP
     (UAP_MAJORAID          ASC,
      UAP_MINORAID          ASC)
    USING STOGROUP SOLASTG
    PRIQTY 7200 SECQTY 1440
    ERASE  NO
    FREEPAGE 4 PCTFREE 25
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLUAR
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLUAR
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 36000 SECQTY 7200
    ERASE  NO
    FREEPAGE 4 PCTFREE 25
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 32
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLUAR In DBXML003.TSXMLUAR
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLUAR
     (UAR_MAJORAID         CHAR(4) FOR SBCS DATA NOT NULL,
      UAR_MINORAID         CHAR(3) FOR SBCS DATA NOT NULL,
      UAR_ROWNUMBR         INTEGER NOT NULL,
      UAR_SOAPXMLI         VARCHAR(3700)  FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLUAR
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLUAR On SOLA600P.TBXMLUAR
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLUAR
    ON SOLA600P.TBXMLUAR
     (UAR_MAJORAID          ASC,
      UAR_MINORAID          ASC,
      UAR_ROWNUMBR          ASC)
    USING STOGROUP SOLASTG
    PRIQTY 7200 SECQTY 1440
    ERASE  NO
    FREEPAGE 4 PCTFREE 25
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLUSR
------------------------------------------------------------------------
--
  CREATE TABLESPACE TSXMLUSR
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS YES
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLUSR In DBXML003.TSXMLUSR
------------------------------------------------------------------------
--
  CREATE TABLE SOLA600P.TBXMLUSR
     (ID                   TIMESTAMP NOT NULL WITH DEFAULT,
      ENVIRONID            CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      GROUPID              CHAR(26) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      USERIDENTITY         CHAR(128) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PUBLICKEYNM          CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      PRIVATEKEYNM         CHAR(64) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      EFFECTIVE            TIMESTAMP NOT NULL WITH DEFAULT,
      EXPIRES              TIMESTAMP NOT NULL WITH DEFAULT,
      DETAIL               VARCHAR(3700) FOR SBCS DATA NOT NULL
        WITH DEFAULT)
    IN DBXML003.TSXMLUSR
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLUSR On SOLA600P.TBXMLUSR
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLUSR
    ON SOLA600P.TBXMLUSR
     (ID                    ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X2XMLUSR On SOLA600P.TBXMLUSR
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X2XMLUSR
    ON SOLA600P.TBXMLUSR
     (USERIDENTITY          ASC,
      ENVIRONID             ASC,
      EXPIRES               ASC)
    USING STOGROUP SOLASTG
    PRIQTY 720 SECQTY 720
    ERASE  NO
    FREEPAGE 0 PCTFREE 0
    GBPCACHE CHANGED
    NOT CLUSTER
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
------------------------------------------------------------------------
-- Database=DBXML003   Stogroup=SOLASTG
-- Tablespace=DBXML003.TSXMLWGT
------------------------------------------------------------------------
--
```

```
  CREATE TABLESPACE TSXMLWGT
    IN DBXML003
    USING STOGROUP SOLASTG
    PRIQTY 480 SECQTY 480
    ERASE  NO
    FREEPAGE 3 PCTFREE 10
    GBPCACHE CHANGED
    TRACKMOD YES
    SEGSIZE 64
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX 0
    CLOSE YES
    COMPRESS NO
    CCSID      EBCDIC
    DEFINE YES
    MAXROWS 255;
--
------------------------------------------------------------------------
--    Table=SOLA600P.TBXMLWGT In DBXML003.TSXMLWGT
------------------------------------------------------------------------
--
  SET CURRENT SQLID='SOLA600P';
--
  CREATE TABLE SOLA600P.TBXMLWGT
     (WIDGET_NUM           CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      WIDGET_COLOR         CHAR(6) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      WIDGET_SIZE          CHAR(1) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      WIDGET_PRICE         DECIMAL(7, 2) NOT NULL WITH DEFAULT,
      WIDGET_SUPPLIER      CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      WIDGET_DESC          CHAR(20) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      WIDGET_MANU_PLANT    CHAR(8) FOR SBCS DATA NOT NULL
        WITH DEFAULT,
      WIDGET_MANU_COST     DECIMAL(7, 2) NOT NULL WITH DEFAULT,
      WIDGET_LEAD_TIME     SMALLINT NOT NULL WITH DEFAULT)
    IN DBXML003.TSXMLWGT
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC
    NOT VOLATILE;
--
------------------------------------------------------------------------
-- Database=DBXML003
--    Index=SOLA600P.X1XMLWGT On SOLA600P.TBXMLWGT
------------------------------------------------------------------------
--
  CREATE UNIQUE INDEX SOLA600P.X1XMLWGT
    ON SOLA600P.TBXMLWGT
     (WIDGET_NUM            ASC)
    USING STOGROUP SOLASTG
    PRIQTY 48 SECQTY 48
    ERASE  NO
    FREEPAGE 10 PCTFREE 10
    GBPCACHE CHANGED
    CLUSTER
    BUFFERPOOL BP0
    CLOSE YES
    COPY NO
    DEFINE YES
    PIECESIZE 2 G;
--
  CREATE ALIAS SOLA600P.V1XMLACC FOR SOLA600P.TBXMLACC ;
--
  CREATE ALIAS SOLA600P.V1XMLALT FOR SOLA600P.TBXMLALT ;
--
  CREATE ALIAS SOLA600P.V1XMLASN FOR SOLA600P.TBXMLASN ;
--
  CREATE ALIAS SOLA600P.V1XMLCER FOR SOLA600P.TBXMLCER ;
--
  CREATE ALIAS SOLA600P.V1XMLCOM FOR SOLA600P.TBXMLCOM ;
--
  CREATE ALIAS SOLA600P.V1XMLENV FOR SOLA600P.TBXMLENV ;
--
  CREATE ALIAS SOLA600P.V1XMLEXT FOR SOLA600P.TBXMLEXT ;
--
  CREATE ALIAS SOLA600P.V1XMLGRP FOR SOLA600P.TBXMLGRP ;
--
  CREATE ALIAS SOLA600P.V1XMLIPA FOR SOLA600P.TBXMLIPA ;
--
  CREATE ALIAS SOLA600P.V1XMLLOG FOR SOLA600P.TBXMLLOG ;
--
  CREATE ALIAS SOLA600P.V1XMLMAP FOR SOLA600P.TBXMLMAP ;
--
  CREATE ALIAS SOLA600P.V1XMLMON FOR SOLA600P.TBXMLMON ;
--
  CREATE ALIAS SOLA600P.V1XMLMSK FOR SOLA600P.TBXMLMSK ;
--
  CREATE ALIAS SOLA600P.V1XMLMTD FOR SOLA600P.TBXMLMTD ;
--
  CREATE ALIAS SOLA600P.V1XMLMTL FOR SOLA600P.TBXMLMTL ;
--
  CREATE ALIAS SOLA600P.V1XMLMTS FOR SOLA600P.TBXMLMTS ;
--
  CREATE ALIAS SOLA600P.V1XMLOFT FOR SOLA600P.TBXMLOFT ;
--
  CREATE ALIAS SOLA600P.V1XMLPGM FOR SOLA600P.TBXMLPGM ;
--
  CREATE ALIAS SOLA600P.V1XMLPOL FOR SOLA600P.TBXMLPOL ;
--
  CREATE ALIAS SOLA600P.V1XMLPRJ FOR SOLA600P.TBXMLPRJ ;
--
  CREATE ALIAS SOLA600P.V1XMLSCH FOR SOLA600P.TBXMLSCH ;
--
  CREATE ALIAS SOLA600P.V1XMLSPT FOR SOLA600P.TBXMLSPT ;
--
```

```
  CREATE ALIAS SOLA600P.V1XMLTOR FOR SOLA600P.TBXMLTOR ;
--
  CREATE ALIAS SOLA600P.V1XMLUAC FOR SOLA600P.TBXMLUAC ;
--
  CREATE ALIAS SOLA600P.V1XMLUAP FOR SOLA600P.TBXMLUAP ;
--
  CREATE ALIAS SOLA600P.V1XMLUAR FOR SOLA600P.TBXMLUAR ;
--
  CREATE ALIAS SOLA600P.V1XMLUSR FOR SOLA600P.TBXMLUSR ;
--
  CREATE ALIAS SOLA600P.V1XMLWGT FOR SOLA600P.TBXMLWGT ;
--
------------------------------------------------------------------------
--    View=SOLA600P.V1XMLUAM
------------------------------------------------------------------------
--
  CREATE VIEW SOLA600P.V1XMLUAM AS
    SELECT UAC_MAJORAID AS UAM_MAJORAID, UAC_MINORAID AS UAM_MINORAID,
           UAC_ROWNUMBR AS UAM_ROWNUMBR, UAC_USERNAME AS UAM_USERNAME,
           UAC_USERDATE AS UAM_USERDATE, UAC_USERTIME AS UAM_USERTIME,
           UAC_ENDPOINT AS UAM_ENDPOINT,
           COALESCE(UAR_SOAPXMLI, VARCHAR(' ', 3700)) AS UAM_SOAPXMLI
      FROM SOLA600P.V1XMLUAC UAC
           LEFT OUTER JOIN
           SOLA600P.V1XMLUAR UAR
           ON(UAC_MAJORAID = UAR_MAJORAID) AND (UAC_MINORAID =
           UAR_MINORAID) AND (UAC_ROWNUMBR = UAR_ROWNUMBR) ;
```

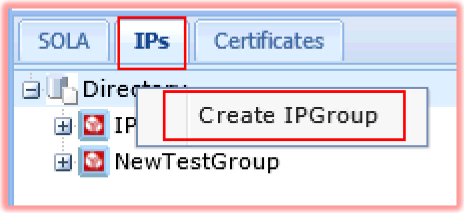
Appendix B: SOLA Custom Channel Lockdown Security

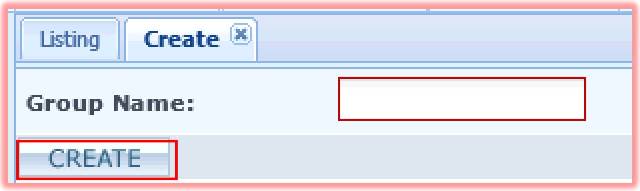
This SOLA custom security feature targets and locks down the client/requestor's IP address and the legacy program that will host the requested Web Service. Channel lockdown security is applied at the Container group level. That is to say that, once specified, the security matrix will apply to all SOLA Container listener containers contained within a listener group.

Creating and Managing IP Addresses

IP Groups

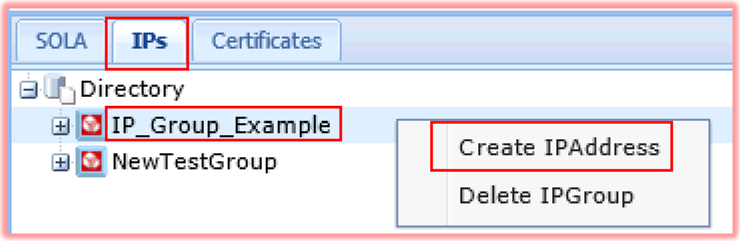
IP groups are used to organize IP addresses and can make creating IP accesses easier. To create a new IP group, right click on the 'Directory' under IP tab and select 'Create IP Group' as illustrated below.

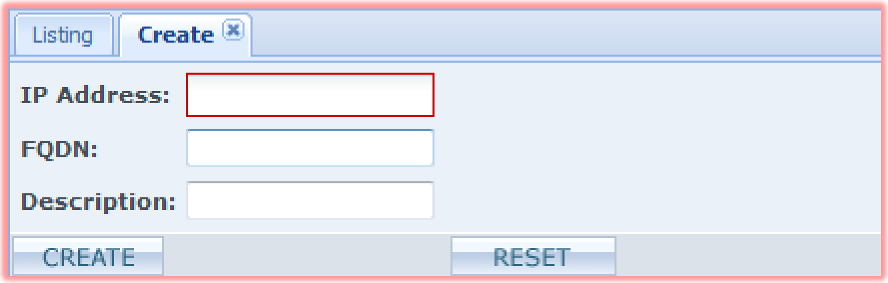




To create a new IP Address, right click the directory root and select **Create IPAddress** from the menu.

This will display the Create tab and allow you to create a new IP address.



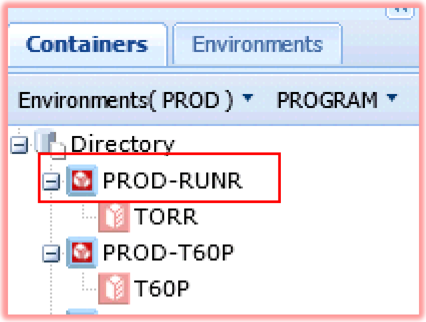


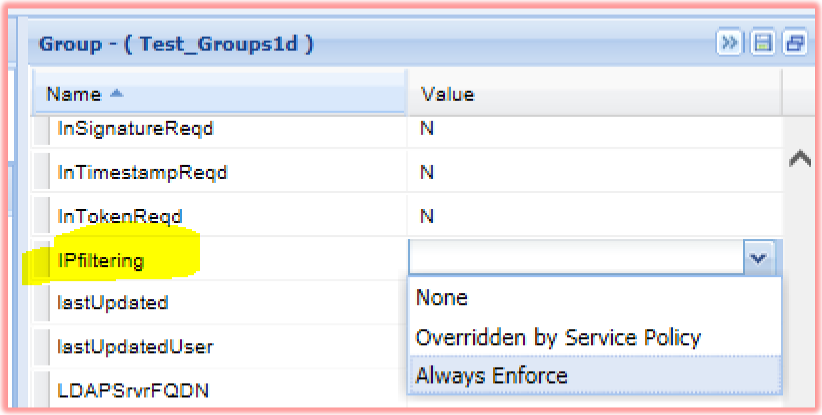
To create an IP address, fill in the following (only the **IP Address** field is required):

- **IP Address:** the IP address (xxx.xxx.xxx.xxx) that you want to authorize. This will typically be the IP address of a client machine that wants to access SOLA web services.
- **FQDN:** the fully qualified domain name
- **Description:** an optional description of the IP address (e.g. "Customer Terminal 4").

You can drag an IP address from one group and drop it on another group to move the IP address.

To activate IP Filtering on Container, set the Container group Property 'IPFiltering' as illustrated below and click the middle icon to save changes.

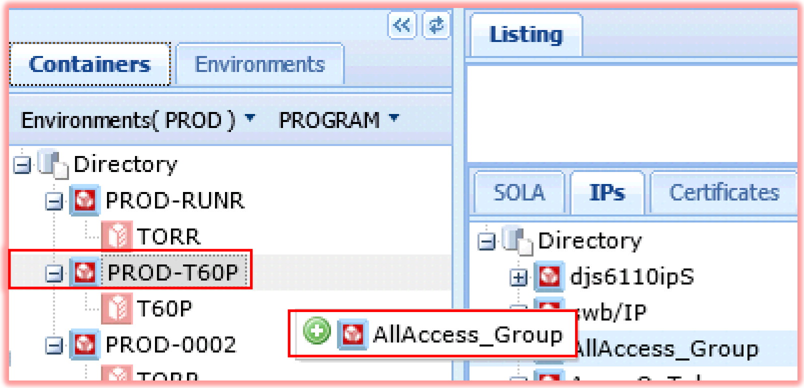




Name ▲	Value
InSignatureReqd	N
InTimestampReqd	N
InTokenReqd	N
IPfiltering	
lastUpdated	
lastUpdatedUser	
LDAPSrvrFQDN	

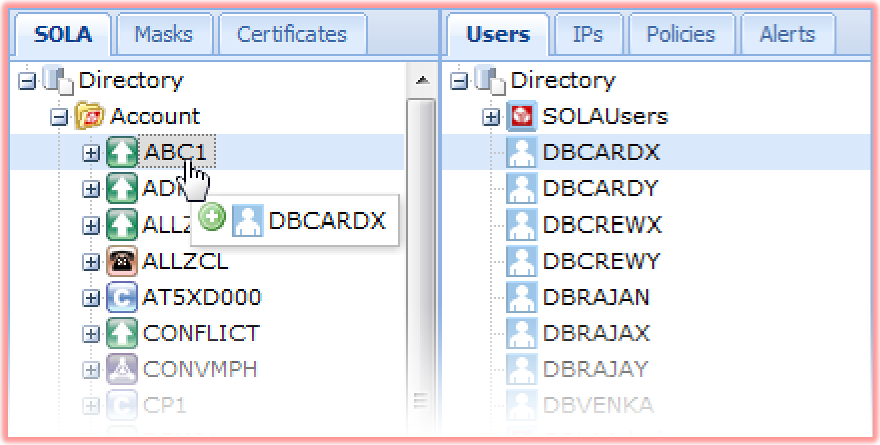
Now the container is enabled for IP Filtering. Drag/Drop IP Groups to the target container group to enable IP addresses of clients that are authorized to send requests to the regions under this group.

NOTE: SOLA supports "IPFiltering" that can be combined with service policy to be enforced as an add-on policy.



Creating an Access

Once you have populated the users and/or IP addresses tab, you can start to create accesses. An access is created when a subject (user or IP) is associated with a resource. This association is accomplished using Resource Manager’s drag and drop capabilities; a subject is dragged over to a resource and dropped into it.



In the following illustration, user DBCARDX is being dragged to program ABC1. The resulting access, once it is deployed, will allow user DBCARDX to invoke that program.

The following table illustrates the effects of various associations:

Subject	Resource	Result
User or User Group	Program	The user or group of users can invoke any method in the target program.
User or User Group	Method	The user or group of users can invoke the target method.
IP or IP Group	Program	All requests coming from the IP or any IP in the IP group can invoke any method in the target program.
IP or IP Group	Method	All requests coming from the IP or any IP in the IP group can invoke the target method.

Once an access is created, you can view it by right clicking on a resource and selecting **Show Accesses**.

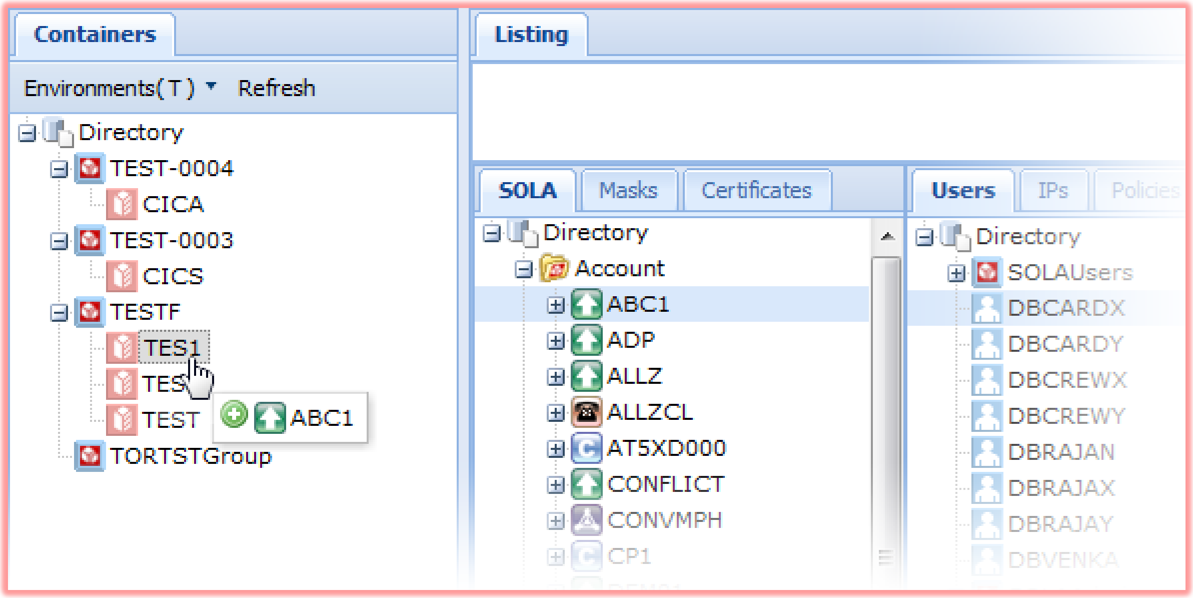
Deploying an Access

When an access is created, it is not active until it is deployed. For example, if you drag a user from the subjects panel to a program in the resource panel, the access you create is not in effect; that user does not have invoke rights to the program. To activate the access, you must deploy that access into a SOLA container group.

The activation of an access is accomplished using Resource Manager's drag and drop capabilities; an access is dragged to a SOLA container, activating that access for every container in the target container's group.

NOTE: although Resource Manager does not allow accesses to be dragged into container groups, an access can only be activated for a container group, not individual containers. Dragging an access into a container activates the access for every container in that container's parent group.

The following illustration shows an access being deployed in the TESTF container group. It could have been dragged to any container in the TESTF group with the same results.



Appendix C: AT-TLS Sample Configuration Data

A1.1 Sample Pagent Configuration

```
#
#  IBM Communications Server for z/OS
#  SMP/E distribution path: /usr/lpp/tcpip/samples/IBM/EZAPAGCO
#
#  Licensed Materials - Property of IBM
#  5694-A01
#  (C) Copyright IBM Corp. 1998, 2005
#  Status = CSV1R7
#
#    PAGENT policy configuration (this file)
#       /usr/lpp/tcpip/samples/pagent.conf
#
#    CommonIpSecConfig policy configuration
#       /usr/lpp/tcpip/samples/pagent_CommonIPSec.conf
#
#    IpSecConfig policy configuration
#       /usr/lpp/tcpip/samples/pagent_IPSec.conf
#
#    TTLSConfig policy configuration
#       /usr/lpp/tcpip/samples/pagent_TTLS.conf
#
#  This file contains sample policy control statements for the Policy Agent
#  which verifies and installs them down to the appropriate MVS
#  TCP/IP stack.  The search order used by the Policy Agent to locate
#  the initial configuration file is (highest priority listed first):
#
#  1) HFS file or MVS data set specified by the -c startup option.  The
#     syntax for an HFS file is '/dir/file' and the syntax for an MVS
#     data set is "//'MVS.DATASET.NAME'".
#  2) HFS file or MVS data set specified with the PAGENT_CONFIG_FILE
#     environment variable.
#  3) /etc/pagent.conf
#
#
#  The following are general rules to be followed when defining policies:
#
#  - Policy Agent configuration files should be specified using code page
#    IBM-1047 for EBCDIC.
#  - Only one attribute and its values can be specified per line.
#  - Text beyond the specified attribute and value is ignored.
#  - Text beginning with the '#' character is a comment and is ignored.
#    Note that comments beginning with the '#' character in an LDAP server
#    'ldif' configuration file may only be recognized as comments at the
#    beginning of the file, therefore such comments should not be specified
#    elsewhere in the file.
#  - For most range specifications, the ranges may be delimited by a
#    colon, a dash, or a blank, but these cannot be mixed within a single
#    range specification.  IP address ranges cannot use the colon delimiter.
#  - The maximum decimal value for numeric values is 4294967295, unless
#    otherwise noted.
#  - IPv6 addresses specified as IPv4-mapped or IPv4-compatible addresses
#    are not valid.
#  - IPv6 policy is installed but is not enforceable in a stack that
#    is not IPv6 enabled
#  - Policy rule and action names are limited to 32 characters.  If QoS
#    and IDS statement names longer than 32 are specified, they are
#    silently truncated.  All other statement names greater than 32
#    will cause an object error (ie.  OBJERR).
#  - For IDS and QoS policy object names that are duplicates,
#    Policy Agent keeps the first entry.
#  - For IPSec and AT-TLS policy object names that are duplicates,
#    Policy Agent keeps the last entry.
#  - Policy objects definedÿin the configuration fileÿand LDAPÿof the
#    same type (ie.  QoS or IDS) with duplicate namesÿare discarded ÿby
#    the Policy Agent and a warning is written to the log file.
#  - Most attributes in configuration files should be specified only once
#    per rule or action (exceptions are noted where appropriate).  When
#    multiple attributes are specified, no error or warning messages are
#    written to the log, and the last instance of the attribute is used.
#  - Attributes for policies defined on an LDAP server may be single- or
#    multi-valued (meaning single or multiple values are allowed for that
#    attribute.  The Policy Agent detects multiple values for attributes
#    that are defined as single-valued, and treats the policy object as
#    in error.
#  - The policy version is specified by the configuration file statement
#    name or the LDAP object class, as follows: "ServicePolicyRules" and
#    "ServiceCategories" statements/object classes specify version 1
#    policies.  "PolicyRule" and "PolicyAction" statements/object classes
#    specify version 2 and up policies.
#  - Errors detected in a policy rule or action result in that policy
#    object being discarded.
#  - If a rule refers to an action that does not exist (or has been discarded
#    due to an error) then the rule is also discarded, if the policy version
#    is 2 or greater.
#
#
#  The following conventions will be used in this configuration file:
#         p  : choose one in the allowed parameter set
#         p+ : choose one or more in the allowed parameter set
#         B  : integer value of a byte (i.e., 0 =< B =< 255)
#         b  : bit string starting with left most bit (e.g., 101 is
#              equivalent 10100000 in a byte field)
#         i  : integer value
#         s  : a character string
#         a  : IPv4 address in dotted-decimal format or
#              IPv6 address in colon-hex format
#         l  : a distinguished name in directory format k=s,k=s...,
#              where k & s are strings
#        (R) : Required parameter
#        (C) : Conditionally required parameter (required if ...)
#        (O) : Optional parameter


# LogLevel Statement
#   This statement specifies what type of log messages should be logged
#   into the Policy Agent log file.  The default is log level 31.  If
#   upon invocation of Pagent, the debug (-d) option is specified with
#   debug level 1, all log messages are logged.
#
#   statement format:
#         LogLevel                                 i  # Logging level.
#   where:
#         i                                     (R):  The sum of the following values that
#                                                      represent log levels:
#                                                      LOGL_SYSERR     1
#                                                      LOGL_OBJERR     2
#                                                      LOGL_PROTERR    4
#                                                      LOGL_WARNING    8
#                                                      LOGL_EVENT     16
#                                                      LOGL_ACTION    32
#                                                      LOGL_INFO      64
#                                                      LOGL_ACNTING  128
#                                                      LOGL_TRACE    256
#
#   example:  LogLevel 15     specifies four error types to be logged:
#                                syserr, objerr, proterr, and warning.


# TcpImage and PEPInstance Statements (synonyms)
#   This statement specifies an MVS TCP/IP image/stack and its associated
#   policy control file to be installed to that image.  If policy control
#   file is not specified, following control statements (if any) in this
#   file will be installed to that image.  If no TcpImage statement is
#   specified, all policies will be installed to the default TCP/IP image.
#   Parameter FLUSH or NOFLUSH (default) can be used to force flushing
#   (deletion) of all existing policy control data in the stack on
#   startup or when the configuration files change.  The PURGE or NOPURGE
#   parameter controls whether or not policies are deleted from the stack
#   when Pagent is shut down. The time interval for checking for new,
#   changed, or deleted policies can be specified.  The default is 1800
#   seconds (30 minutes).
#
#   statement format:
#         TcpImage | PEPInstance     s1   s2   p   p   i  # TCP/IP images pecification.
#   where:
#         s1                                    (R):  (8 characters) is the name of the MVS
#                                                      TCP/IP image.
#         s2                                    (O):  Is the path of the policy control file.
#                                                      If not specified, this file is used.
#         p                                     (O):  FLUSH | NOFLUSH, default is NOFLUSH.
#         p                                     (O):  PURGE | NOPURGE, default is NOPURGE.
#         i                                     (O):  File/LDAP modification check interval in
#                                                      seconds.  Default is 1800 (30 minutes).
#
#   example:  TcpImage TCPCS /tmp/TCPCS.policy  FLUSH PURGE 600

# CommonIpSecConfig statement
#   The CommonIpSecConfig statement specifies the path of an IPSec
#   policy file that contains common IPSec policy statements.  These
#   common statements can be referenced from a stack specific IPSec
#   policy file.  To define a common set of policies for multiple
#   stacks, the IpSecConfig statement can specify the same
#   file as the CommonIpSecConfig statement.
#
#   Stack specific IPSec policies are defined in a stack IPSec
#   specific policy file.  A stack specific IPSec policy file is
#   identified by an IpSecConfig statement.
#
#   The refresh interval for the CommonIpSecConfig file is
#   inherited from the main configuration file.
#
#   The CommonIpSecConfig statement may only appear in the main
#   configuration file.
#
#   If a CommonIpSecConfig statement appears multiple times in the
#   main configuration file the last occurrence of the statement
#   will be used.  If the CommonIpSecConfig statement appears in an
#   image configuration file it is ignored.
#
#   The configuration information defined in the file identified with
#   the CommonIpSecConfig statement is prepended to the
#   configuration information defined in files identified with the
#   IpSecConfig statement.  There are 2 consequences to this:
#
#   - If no IpSecConfig statements are specified, then the
#     CommonIpSecConfig file is not parsed by Policy Agent.  The
#     IpSecConfig statement is required to define IPSec policy for a
#     given stack.
#   - If multiple stacks are defined, the CommonIpSecConfig file is
#     parsed for each stack, so any errors contained in the file are
#     reported multiple times.
#
#   statement format:
#         CommonIPSecConfig    s1
#   where:
#         s1                                    (R):  The path of the common
#                                                     IPSec policy file to be
#                                                     installed.
#
#   example:
#     CommonIPSecConfig /usr/lpp/tcpip/samples/pagent_CommonIPSec.conf

# IPSecConfig Statements
#   The IpSecConfig statement specifies the path of an IPSec
#   policy file that contains stack specific IPSec policy statements.
#   The IpSecConfig statement is required to define IPSec policy
#   for a given stack.  To define a common set of policies for multiple
#   stacks, the IpSecConfig statement can be specified with no
#   path name.
#
#   The refresh interval for the IpSecConfig file is inherited
#   from the image configuration file containing the corresponding
#   IpSecConfig statement.
#
#   The IpSecConfig statement may only appear in an image
#   configuration file.  If an IpSecConfig statement appears
#   multiple times in an image configuration file the last occurrence of
#   the statement will be used.  If the IpSecConfig statement
#   appears in the main configuration file it is ignored.
#
#   statement format:
#         IPSecConfig    s1
#   where:
#         s1                                    (O):  The path of the stack
#                                                     specific IPSec policy
#                                                     file to be installed.
#                                                     If no path name is
#                                                     specified, then the
#                                                     common IPSec policy file
#                                                     specified on the
#                                                     CommonIpSecConfig
#                                                     statement is used.
#
#   example:
#     IPSecConfig /usr/lpp/tcpip/samples/pagent_IPSec.conf

# CommonTTLSConfig statement
#   The CommonTTLSConfig statement specifies the path of a TTLS
#   policy file that contains common TTLS policy statements.  These
#   common statements can be referenced from a stack specific TTLS
#   policy file.  To define a common set of policies for multiple
#   stacks, the TTLSConfig statement can specify the same
#   file as the CommonTTLSConfig statement.
#
#   Stack specific TTLS policies are defined in a stack TTLS
#   specific policy file.  A stack specific TTLS policy file is
#   identified by a TTLSConfig statement.
#
#   The refresh interval for the CommonTTLSConfig file is
#   inherited from the main configuration file.
#
#   The CommonTTLSConfig statement may only appear in the main
#   configuration file.
#
#   If a CommonTTLSConfig statement appears multiple times in the
#   main configuration file the last occurrence of the statement
#   will be used.  If the CommonTTLSConfig statement appears in an
#   image configuration file it is ignored.
#
#   The configuration information defined in the file identified with
#   the CommonTTLSConfig statement is prepended to the
#   configuration information defined in files identified with the
#   TTLSConfig statement.  There are 2 consequences to this:
#
#   - If no TTLSConfig statements are specified, then the
#     CommonTTLSConfig file is not parsed by Policy Agent.  The
#     TTLSConfig statement is required to define TTLS policy for a
#     given stack.
#   - If multiple stacks are defined, the CommonTTLSConfig file is
#     parsed for each stack, so any errors contained in the file are
#     reported multiple times.
#
#   statement format:
#         CommonTTLSConfig    s1
#   where:
#         s1                                    (R):  The path of the common
#                                                     TTLS policy file to be
#                                                     installed.
#
#   example:
#     CommonTTLSConfig /usr/lpp/tcpip/samples/pagent_TTLS.conf

# TTLSConfig Statements
#   The TTLSConfig statement specifies the path of a TTLS
#   policy file that contains stack specific TTLS policy statements.
#   The TTLSConfig statement is required to define TTLS policy
#   for a given stack.  To define a common set of policies for multiple
#   stacks, the TTLSConfig statement can be specified with no
#   path name.
#
#   The refresh interval for the TTLSConfig file is inherited
#   from the image configuration file containing the corresponding
#   TTLSConfig statement.
#
#   The TTLSConfig statement may only appear in an image
#   configuration file.  If a TTLSConfig statement appears
#   multiple times in an image configuration file the last occurrence of
#   the statement will be used.  If the TTLSConfig statement
#   appears in the main configuration file it is ignored.
```

```
#
#   statement format:
#         TTLSConfig    s1   p    p
#   where:
#         s1                                     (O):  The path of the stack
#                                                      specific TTLS policy
#                                                      file to be installed.
#                                                      If no path name is
#                                                      specified, then the
#                                                      common TTLS policy file
#                                                      specified on the
#                                                      CommonTTLSConfig
#                                                      statement is used.
#         p                                      (O):  FLUSH | NOFLUSH,
#                                                      default value is
#                                                      obtained from the
#                                                      corresponding TcpImage or
#                                                      PEPInstance statement.
#         p                                      (O):  PURGE | NOPURGE,
#                                                      default value is
#                                                      obtained from the
#                                                      corresponding TcpImage or
#                                                      PEPInstance statement.
#
#   example:
#     TTLSConfig /usr/lpp/tcpip/samples/pagent_TTLS.conf FLUSH PURGE
      TTLSConfig /etc/pagent_TTLS.conf FLUSH PURGE

# ReadFromDirectory Statement
#   This statement initializes the LDAP client so that the rules will be
#   downloaded from the LDAP server in addition to being read from this
#   configuration file.
#
#   statement format:
#         ReadFromDirectory
#         {
#            LDAP_Server                        (a|s)  # Name or IPv4 address of the directory server.
#            LDAP_Port                              i  # The port of the directory server.
#            LDAP_BackupServer                  (a|s)  # Name or IPv4 address of the backup directory
#                                                      #  server.
#            LDAP_BackupPort                        i  # The port of the backup directory server.
#            LDAP_DistinguishedName                 l  # LDAP logon id.
#            LDAP_Password                          s  # LDAP logon password.
#            LDAP_SSL                                  # LDAP SSL security specification.
#            {
#               LDAP_SSLKeyringFile                 s  # SSL key ring file specification.
#               LDAP_SSLKeyringPassword             s  # Password to the key ring file.
#               LDAP_SSLName                        s  # Key ring label name.
#            }
#            LDAP_SessionPersistent                 p  # Should the LDAP session with the server
#                                                      #  be kept open?
#            LDAP_ProtocolVersion                 2|3  # LDAP protocol version.
#            LDAP_SchemaVersion                 1|2|3  # Policy schema version.
#            Base                                   l  # The base to look up policies from the server
#                                                      #  (for version 1 policies).
#            LDAP_SelectedTag                       s  # A tag to select policies for this host
#                                                      #  (for version 1 policies).
#            LDAP_AbstractPolicy                    p  # Whether or no abstract policy searching is
#                                                      #  supported by the LDAP server.
#            SearchPolicyBaseDN                     l  # The base to look up policies from the server
#                                                      #  (for version 2 and up policies).
#            SearchPolicyKeyword                    s  # Search keyword for policy objects
#                                                     #  (for version 3 policies).
#            SearchPolicyGroupKeyWord              s  # Search keyword for policy group objects
#                                                     #  (for version 2 and up policies).
#            SearchPolicyRuleKeyWord               s  # Search keyword for policy rule objects
#                                                     #  (for version 2 and up policies).
#            PolicyRole                            s  # Roles or role-combinations played by this
#                                                     #  LDAP client (i.e. Policy Agent).
#         }
#   where:
#         LDAP_Server                           (O):  LDAP_Server is the domain name or the IPv4
#                                                      address of the directory server. If the user
#                                                      does not specify this value, the default
#                                                      is the local host which is 127.0.0.1.
#
#         LDAP_Port                             (O):  LDAP_Port is the port on which the
#                                                      directory server is running.  If the user
#                                                      does not specify this value, the default
#                                                      LDAP port of 389 is used.
#
#         LDAP_BackupServer                     (O):  Domain name or IPv4 address of the
#                                                      backup directory server.  This is used
#                                                      when Pagent can't contact the primary
#                                                      server.  Default is no backup server.
#
#         LDAP_BackupPort                       (O):  The port on which the backup directory server
#                                                      is running.   Default is 389.
#
#         LDAP_DistinguishedName                (C):  The Distinguished Name for userid to be
#                                                      used when connecting to the LDAP server.
#                                                      If this attribute is not specified,
#                                                      anonymous userid is used for the connect.
#                                                      If this attribute is specified,
#                                                      LDAP_Password must also be specified.  Case
#                                                      sensitivity is determined by the LDAP server.
#
#         LDAP_Password                         (C):  The password to be used when connecting to
#                                                      the LDAP server.  If this attribute is
#                                                      specified.  LDAP_DistinguishedName must
#                                                      also be specified.
#
#         LDAP_SSLKeyringFile                   (C):  The name of the keyring file, which
#                                                      contains the certificates trusted by the
#                                                      client.  The file may also contain a
#                                                      public key and certificate.  This parameter
#                                                      is required if LDAP_SSL is specified.
#
#         LDAP_SSLKeyringPassword               (O):  The password of the keyring file.  The
#                                                      password is set using the gskkyman tool.
#
#         LDAP_SSLName                          (O):  The label assigned to your private key /
#                                                      certificate pair, created with the gskkyman
#                                                      tool.
#
#         LDAP_SessionPersistent                (O):  LDAP_SessionPersistent is YES|NO that
#                                                      indicates if the session with the LDAP
#                                                      server should be kept open or not, for the
#                                                      purpose of querying for updates at the
#                                                      interval specified on the TcpImage statement.
#                                                      If this interval is small, the value of
#                                                      keeping the session opened is greater, to
#                                                      reduce the overhead of opening the session
#                                                      for each query.  The default is NO.
```

```
#
#        LDAP_ProtocolVersion                   (O):  LDAP protocol version to be used.  Supported
#                                                      version is 2 or 3.  Default is 3.
#
#        LDAP_SchemaVersion                     (O):  LDAP Policy schema version.  Version 1 is for
#                                                      policy schemas from releases prior to CS
#                                                      for OS390 V2R10.  Version 2 is for policy
#                                                      schemas for CS for OS390 V2R10.  Version 3
#                                                      is for policy schemas as of OS390 V1R2.
#                                                      Default is 3.
#
#        Base                                   (C):  Base is the distinguished name of the
#                                                      subtree in the directory where the policies
#                                                      are located.  This parameter is required
#                                                      with schema version 1.
#
#        LDAP_SelectedTag                       (O):  LDAP_SelectedTag is any string which can
#                                                      be used to select a subset of the policies
#                                                      under the base tree.  If this value is not
#                                                      specified, the first name returned by
#                                                      gethostname() is used as the tag.  This
#                                                      parameter is used in searching version 1
#                                                      schema.
#
#        LDAP_AbstractPolicy                    (O):  LDAP_AbstractPolicy is YES|NO.  Choose YES
#                                                      for LDAP version 3 servers that are
#                                                      capable of matching objectClass for abstract
#                                                      and auxiliary classes.  Choose NO otherwise.
#                                                      When YES is chosen, Pagent uses
#                                                      objectClass=ibm-policy when searching the
#                                                      server.  Otherwise, it uses objectClass=*
#                                                      (all object classes).  The default is YES.
#
#        SearchPolicyBaseDN                     (C):  The distinguished name of the subtree under
#                                                      which to find policies that are defined
#                                                      with schema version 2 and up.  This parameter
#                                                      is required with schema version 2 and up.
#                                                      Case sensitivity is determined by the LDAP
#                                                      server.
#
#        SearchPolicyKeyword                    (O):  Keyword used to search for policy objects
#                                                      under the subtree.  This is only allowed
#                                                      with version 3 schema and it is used in
#                                                      the initial search.  Up to 8 instances of
#                                                      this attribute can be specified.  This value
#                                                      is matched against the policyKeywords
#                                                      attribute in the policy rules.  Case
#                                                      sensitivity is determined by the LDAP server.
#
#        SearchPolicyGroupKeyWord               (O):  Keyword used to search for policy groups under
#                                                      the subtree (e.g., search scoping).  This is
#                                                      only allowed with version 2 and up schema and
#                                                      it is used in the initial search.  Up to 8
#                                                      instances of this attribute can be specified.
#                                                      This value is matched against the
#                                                      policyGroupKeywords attribute in the policy
#                                                      groups.  Case sensitivity is determined
#                                                      by the LDAP server.
#
#        SearchPolicyRuleKeyWord                (O):  Keyword used to search for policy rules under
#                                                      the subtree.  This is only allowed with
#                                                      version 2 and up schema and it is used in the
#                                                      initial search.  Up to 8 instances of this
#                                                      attribute can be specified.  This value is
#                                                      matched against the policyRuleKeywords
#                                                      attribute in the policy rules.  Case
#                                                      sensitivity is determined by the LDAP server.
#
#         PolicyRole                              (O):  Specifies a policy role or role-combination
#                                                       (see below).  Use this parameter to filter
#                                                      the policy rules to be retrieved.  This
#                                                      parameter is only valid with schema version
#                                                      3.  This parameter can be repeated as many
#                                                      times as necessary.  Either a single role or
#                                                      a set of roles, known as a role-combination,
#                                                      may be specified.  The roles may be single
#                                                      words, or any strings containing blanks or
#                                                      other special characters, contained in
#                                                      double quotes.  Role-combinations are
#                                                      specified as follows.  The first role is
#                                                      specified the same way that a single role is
#                                                      specified.  Each additional role in the
#                                                      role-combination is prefixed with the
#                                                      characters "&&".  For example:
#                                                       PolicyRole     role1
#                                                       PolicyRole     &&"quoted role 2"
#                                                       PolicyRole     "quoted role 3"
#                                                       PolicyRole     role4
#                                                      This parameter is used to filter out policy
#                                                      rules that don't contain any of the
#                                                      specified roles or role-combinations, using
#                                                      the attribute ibm-policyRoles.  For example,
#                                                      the set of roles specified above result in
#                                                      the retrieval of any policy rules that
#                                                      specify "role1&&quoted role 2" or "quoted
#                                                      role3" or "role4" in their policyRoles
#                                                      values.
#
# Note: If the LDAP server being used is an z/OS LDAP server prior to   Version 2 Release 10 then
#       only one of the parameters SearchPolicyGroupKeyWord or SearchPolicyRuleKeyWord may be used.
#       If both parameters are used the LDAP server will not return any policy information.
#
#
#   example:  ReadFromDirectory
#             {
#               LDAP_Server             9.10.11.12
#               LDAP_Port               9000
#               LDAP_DistinguishedName  cn=root, o=IBM, c=US
#               LDAP_Password           secret
#               LDAP_ProtocolVersion    2
#               LDAP_SchemaVersion      2
#               SearchPolicyBaseDN      cn=group5, o=IBM, c=US
#             }

# PolicyPerfMonitorForSDR Statement
#   This statement is used to enable/disable (without this statement, this
#   function is turned off) the policy performance monitor function that
#   assigns a QoS weight fraction to the monitored policy performance data and
#   sends messages to the Sysplex Distributor (SD) routing component as the
#   monitored data crosses its defined thresholds.  SD uses this weight
#   fraction to influence its routing decision on the incoming connection
#   requests to appropriate hosts within a group that is responsible for
#   processing the request.   The QoS weight fraction is used by SD to reduce
#   the availability factor that SD obtains from the Work Load Manager (WLM).
#   For instance, assume the WLM weight (available processing capacity) is 64 and
#   the QoS performance monitor detects a significant loss rate over the network
#   from the corresponding node in the sysplex.  Also assume that the QoS
#   weight fraction is 50%: the resulting weight used in routing incoming
#   connection requests to that node is now only 32 (64x0.5) instead of 64.
#   This weight adjustment results in balancing the work load in the sysplex
#   taking into account both the node processing capacity and the QoS
#   performance over the network.
#
#   Note that there must exist at least one policy rule definition that covers
#   the application whose incoming connection request is being routed by SD.
#   The policy rule(s) enables the collection of the performance data.  This
#   function only works with TCP since other IP transports contain data
#   retransmission and error recovery.   Also, if policies with policyScope TR
#   exist that cover the application, and if a limit on the total connections
#   is defined, then when the total active connections for that application
#   reaches the contrained state (i.e. 90% of the defined total), the QOS weight
#   fraction will be set to 100%.   This will divert incoming connection
#   requests for the application away from the corresponding target node.
#
#   statement format:
#         PolicyPerfMonitorForSDR     Enable/Disable  # Default is Disable. If Enable,
#                                                     #  the following parameters can
#                                                     #  be specified.
#         {
#            SamplingInterval                      i  # How often to sample the policy
#                                                     #  performance information.
#            LossRatioAndWeightFr                i i  # The first number is the unit ratio of
#                                                     #  retransmitted bytes (loss) over
#                                                     #  transmitted bytes; the second is
#                                                     #  the weight fraction to be assigned.
#            LossMaxWeightFr                       i  # Maximum weight fraction to be assigned.
#            TimeoutRatioAndWeightFr             i i  # The first number is the unit ratio of
#                                                     #  the number of timeouts over
#                                                     #  total packets transmitted; the second
#                                                     #  is the weight fraction to be assigned.
#            TimeoutMaxWeightFr                    i  # Maximum weight fraction to be assigned.
#         }
#   where:
#         SamplingInterval                     (O):  In units of seconds. Default is 60 seconds.
#         LossRatioAndWeightFr                 (O):  Loss (Retransmission) Ratio is in units of
#                                                     tenths of a percent (1-1000) and the
#                                                     weight fraction is in percentage units
#                                                     (1-100%).  Default is 10 and 10, meaning 1%
#                                                     unit loss ratio corresponds to 10% fraction.
#                                                     A weight fraction of 0 means to suppress
#                                                     loss ratio factor in SD routing.
#         LossMaxWeightFr                      (O):  Weight fraction is added or subtracted
#                                                     as the ratio exceeds the next unit or
#                                                     retreats to the lower unit.  If the
#                                                     monitored loss ratio increases such that
#                                                     the weight fraction exceeds the maximum,
#                                                     only maximum weight fraction is sent to SD.
#                                                     Default is 100, or 100%.
#         TimeoutRatioAndWeightFr              (O):  Time out Ratio is also in units of tenths of
#                                                     a percent and the weight fraction is in
#                                                     percentage.  Default is 10 and 20,
#                                                     respectively.  A weight fraction of 0 means
#                                                     to suppress timeout ratio factor in SD
#                                                     routing.
#         TimeoutMaxWeightFr                   (O):  Default is 100%.
#
```

```
#   example:  PolicyPerfMonitorForSDR    Enable
#             {
#                Samplinginterval          120
#                LossRatioAndWeightFr    10 20
#                LossMaxWeightFr           100
#                TimeoutRatioAndWeightFr  5 50
#                TimeoutMaxWeightFr        100
#             }
#
#      In this example, Pagent will send to SDR a message when the loss
#      (retransmission) ratio begins to exceed 1% but not above 2%, with
#      a weight fraction of 20%  (this means that the WLM weight will be
#      decreased by 20% before it is used as a measure to route incoming
#      connection requests).   When the loss (retransmission) ratio exceeds
#      2% but not above 3%, a message is sent with a weight fraction of 40%,
#      and so on.  When the loss exceeds 5%, a maximum weight fraction of 100%
#      will be used.  The same with the timeout ratio.  When the timeout ratio
#      exceeds 0.5% but not above 1%, a weight fraction of 50%  is added to
#      QoS fraction in the message sent to SD.  And so on.

# PolicyPerformanceCollection Statement
#   This statement is used to enable/disable the policy performance collection
#   function. Without this statement, this function is turned off by default.
#   When enabled, the Policy Agent collects performance data from the kernel
#   and caches it. This performance data is then made available to user
#   applications through the Policy API (PAPI). This data can also be
#   optionally logged to a performance log file for offline performance
#   analysis.
#
#   statement format:
#         PolicyPerformanceCollection Enable/Disable  # Default is Disable.
#         {
#            DataCollection                        p+ # Specifies the type of performance
#                                                     #  data that needs to be collected.
#            MinimumSamplingInterval               i  # The minimum time at which performance
#                                                     #  data will be collected from the kernel
#            LogSamplingInterval                   i  # The time at which performance data will
#                                                     #  be collected from the kernel and logged
#                                                     #  into the performance log file.
#            PerformanceLogFile                    s  # Specifies the name of the file to which
#                                                     #  performance data needs to be logged.
#            NumberOfLogFiles                      i  # Specifies the number of performance log
#                                                     #  files to be maintained.
#            SizeOfLogFile                         i  # Specifies the size of each log file.
#         }
#   where:
#         DataCollection                        (O):  Specifies the type of performance data that
#                                                      will be collected. The values could be
#                                                      RULE or ACTION. Default is RULE.
#         MinimumSamplingInterval               (O):  In seconds. Value ranges from 30 to
#                                                      2147483647. Default is 30.
#         LogSamplingInterval                   (C):  In seconds. Value ranges from 30 to
#                                                      2147483647. If LogSamplingInterval is
#                                                      specified then PerformanceLogFile needs
#                                                      to be specifed.
#         PerformanceLogFile                    (C):  Specifies the name of the log file.
#                                                      PerformanceLogFile needs to be specified
#                                                      if LogSamplingInterval is specified.
#         NumberOfLogFiles                      (O):  Ranges from 1 - 255. Default is 3.
#         SizeOfLogFile                         (O):  In kilobytes (Kb). Value ranges from 1 -
#                                                      1000000Kb.  Default is 300 Kb.
#
```

```
#   example:  PolicyPerformanceCollection   Enable
#             {
#                DataCollection            RULE ACTION
#                MinimumSamplingInterval   60
#                LogSamplingInterval       60
#                PerformanceLogFile        /tmp/perfdata.log
#             }
#
#      In this example, Pagent will collect the performance data from the
#      kernel every 60 seconds and will log the performance data to the
#      performance log file (/tmp/perfdata.log) every 60 seconds.

# SetSubnetPrioTosMask Statement
#   This statement defines the TOS/priority field in the IP header type of
#   service byte.  It is used by the TCP/IP stack to read the TOS value and
#   assign appropriate service to the corresponding IP packets.  If this
#   statement is not specified, TCP/IP will use the system default TOS mask
#   and priority levels for all interfaces currently defined for IPv4
#   (RFC 791).  Note that there is an alternate definition for the TOS
#   byte referred to as the DS (Differentiated Services) byte, see RFC 2474
#   for details.  This statement can be used to support the DS byte format.
#   statement format:
#         SetSubnetPrioTosMask
#         {
#            SubnetAddr                            a  # Subnet IP address.
#            SubnetTosMask                         b  # TOS mask to obtain priority level.
#            PriorityTosMapping                B b B  # Priority level, corresponding TOS,
#                                                     # and optional user priority.
#         }
#   where:
#         SubnetAddr                           (O):  Is the local subnet interface address.
#                                                     Default is 0, meaning the mask is the same
#                                                     for all interfaces.
#         SubnetTosMask                        (R):  The TOS mask  (e.g., 11100000).
#         PriorityTosMapping                   (O):  This key can be repeated for each priority
#                                                     level to Tos value mapping.  For example:
#                                                     PriorityTosMapping  0 0
#                                                     PriorityTosMapping  1 01000000
#
#                                                     The third parameter is the user priority,
#                                                     which is also known as the VLAN priority,
#                                                     and is an optional parameter.
#                                                     This allows the Virtual LAN (VLAN) user
#                                                     priority to be set for those devices
#                                                     that support tagging of VLAN frames.
#
# CAUTION: The coding of the user priority (third parameter) will cause the frame to be sent out
# based on the IEEE 802.1Q specification  which establishes a standard method for tagging Ethernet
# frames with VLAN priority and membership information. Specifically, VLAN priority tagged frame
# is used to convey packet priority to the switches, and has NULL for VLANID. A full VLAN tagged
# frame supported as of CS for z/OS V1R5 contains both the priority and non-NULL VLANID. If you
# have switches in your network that DO NOT support the IEEE 802.1Q standard (either VLAN priority
# tagged frames or full VLAN tagged frames) or are not properly configured for this frame, the
# frame may be dropped by the switch.
#
#   example:  SetSubNetPrioTosMask
#             {
#               SubnetTosMask       11100000
#               PriorityTosMapping  1 1110000 7
#               PriorityTosMapping  1 1100000 6
#               PriorityTosMapping  2 1010000 5
#               PriorityTosMapping  2 1000000 4
#               PriorityTosMapping  3 0110000 3
#               PriorityTosMapping  3 0100000 2
#               PriorityTosMapping  4 0010000 1
#               PriorityTosMapping  4 0000000 0
#             }


# PolicyAction Statement
#   This statement specifies the QoS that a flow of IP packets
#   (e.g., from a TCP connection, or UDP data) should receive end-to-end
#   as they traverse the network.  In addition to QoS, this statement
#   can also be used to specify Traffic Regulation Management action to
#   be performed by TCP/IP for a target application specified in the
#   policyRule.
#
#   statement format:
#         policyAction                         s  # Policy action name.
#         {
#            PolicyScope                       p  # Type of action to be performed.
#            Permission                        p  # If packets belonging to this
#                                                 #  action should be discarded
#                                                 #  or allowed to proceed.
#            MaxRate                           i  # Maximum rate/thoughput per TCP
#                                                 #  connection allowed for traffic
#                                                 #  in this action.
#            MinRate                           i  # Minimum rate/throughput per TCP
#                                                 #  connection allowed for traffic
#                                                 #  in this action.
#            MaxDelay                          i  # Maximum end-to-end delay time.
#            OutgoingTOS                       b  # TOS/DS value of outbound traffic
#                                                 #  belonging to this action.
#            MaxConnections                    i  # Maximum number of end-to-end TCP
#                                                 #  connections at any instance of time
#                                                 #  for this action.
#            OutboundInterface                 a  # Sysplex Distributor routing interface.
#                                                 # The following are leaky bucket traffic
#                                                 #  conditioning parameters:
#            DiffServInProfileRate             i  # Token generation rate.
#            DiffServInProfilePeakRate         i  # Leaky bucket peak rate.
#            DiffServInProfileTokenBucket      i  # Burst size.
#            DiffServInProfileMaxPacketSize    i  # Peak rate max packet size.
#            DiffServOutProfileTransmittedTOSByte  b  # TOS/DS for out-of-profile
#                                                 #  traffic.
#            DiffServExcessTrafficTreatment    p  # Treatment for out-of-profile traffic.
#                                                 # The following are RSVP parameters:
#            InboundScope                      p  # Inbound QoS service type.
#            AverageConnectionRate             i  # Average new inbound connections per
#                                                 # second.
#            ConnectionBurstSize               i  # Maximum number of new concurrent
#                                                 # inbound connections.
#            PeakConnectionRate                i  # Peak rate of inbound connection
#                                                 # token bucket conditioner.
#            AverageApplicationRate            i  # Average new application requests
#                                                 # per second.
#            AverageApplicationRequestRate     i  # Synonym for AverageApplicationRate.
#            ApplicationBurstSize              i  # Maximum number of new concurrent
#                                                 # application requests.
#            ApplicationRequestBurstSize       i  # Synonym for ApplicationBurstSize.
#            ApplicationPeakRate               i  # Peak rate of application request
#                                                 # token bucket conditioner.
#            ApplicationRequestPeakRate        i  # Synonym for ApplicationPeakRate.
#            PrioritizedQueue                  p  # Order of processing inbound
#                                                     # connections.
#            FlowServiceType                       p  # Type of RSVP reserved traffic.
#            MaxRatePerFlow                        i  # Maximum rate a flow in this service
#                                                     #  category is allowed to reserve.
#            MaxTokenBucketPerFlow                 i  # Maximum token bucket size per flow.
#            MaxFlows                              i  # Maximum number of reserved flows.
#            SignalClient                          p  # Enable RSVP for TCP/UDP connection.
#                                                     # The following parameters are for
#                                                     #  TR (Traffic Regulation
#                                                     #  Management) action:
#            TypeActions                          p+  # TR action type.
#            TotalConnections                      i  # Total connections allowed for an
#                                                     #  application specified in a policy
#                                                     #  rule.
#            Percentage                            i  # Percentage of connections allowed.
#            TimeInterval                          i  # TR sampling interval in minutes.
#            LoggingLevel                          i  # TR logging level.
#         }
#   where:
#         s                                     (R):  Is the name of this policy action
#                                                      (up to 32 characters, truncated if
#                                                      longer).
#         PolicyScope                           (O):  DataTraffic | RSVP | Both | TR
#                                                      TR is Traffic Regulation Management.
#                                                      Default is Both, which means both
#                                                      DataTraffic and RSVP.
#         Permission                            (O):  Allowed | Blocked, default is Allowed.
#         MaxRate                               (O):  in Kbps (K bits per second), default is
#                                                      infinite.
#         MinRate                               (O):  In Kbps, default is zero.
#         MaxDelay                              (O):  Default is non-specified (infinite).
#         OutgoingTOS                           (O):  Default is 0.
#         MaxConnections                        (O):  Default is non-specified (infinite).
#         OutboundInterface                     (O):  Sysplex Distributor routing interfaces.
#                                                      Up to 32 instances of this attribute may be
#                                                      specified.  If 0 is specified, then the SD
#                                                      routing component can use any available
#                                                      target server if the target servers
#                                                      identified with instances of this attribute
#                                                      are not available.  Default is no policy
#                                                      control of Sysplex Distributor routing
#                                                      interfaces.  Only IPv4 addresses can
#                                                      be specified.
#         DiffServInProfileRate                 (C):  In Kbps, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         DiffServInProfilePeakRate             (C):  In Kbps, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         DiffServInProfileTokenBucket          (C):  In Kb, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         DiffServInProfileMaxPacketSize        (C):  In Kb, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         DiffServOutProfileTransmittedTOSByte  (O):  See EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         DiffServExcessTrafficTreatment        (O):  Drop | BestEffort | Shape
#                                                      see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         InboundScope                          (O):  Connection | Application
#                                                      see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         AverageConnectionRate                 (O):  In Kbps, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         ConnectionBurstSize                   (O):  In Kb, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         PeakConnectionRate                    (O):  In Kbps, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         AverageApplicationRate
#         AverageApplicationRequestRate         (O):  In Kbps, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         ApplicationBurstSize
#         ApplicationRequestBurstSize           (O):  In Kb, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         ApplicationPeakRate
#         ApplicationRequestPeakRate            (O):  In Kbps, see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         PrioritizedQueue                      (O):  1 | 2 | 3 | 4
#                                                      see EZAPAGAT (pagent_at.conf)
#                                                      for details.
#         FlowServiceType                       (O):  ControlledLoad | Guaranteed, default is
#                                                      ControlledLoad.  Guaranteed is considered
#                                                      greater than ControlledLoad.  Use this
#                                                      attribute to limit the type of RSVP service
#                                                      requested by RSVP applications.
#         MaxRatePerFlow                        (O):  In Kbps, default is system defined
#                                                      maximum.  Use this attribute to limit the
#                                                      mean rate requested by RSVP applications
#                                                      in the traffic specification (Tspec) or
#                                                      Guaranteed service Rspec.
#         MaxTokenBucketPerFlow                 (O):  In Kb, default is system defined maximum.
#                                                      Use this attribute to limit the token bucket
#                                                      depth requested by RSVP applications in the
#                                                      traffic specification (Tspec).
#         MaxFlows                              (O):  Default is non-specified (infinite).
#         SignalClient                          (O):  0 (no signalling) | 1 (signalling)
#         TypeActions                           (C):  Statistics | Log | Limit
#                                                      Required with PolicyScope TR.  Any or all
#                                                      of the values can be specified, for
#                                                      Example: TypeActions Log Limit
#         TotalConnections                      (C):  Required if TypeActions contains Log or Limit.
#         Percentage                            (C):  Required if TypeActions contains Log or Limit.
#         TimeInterval                          (O):  Used when TypeActions is Statistics,
#                                                      default is 60 minutes.
#         LoggingLevel                          (O):  Maximum logging level is 7.  Default is 0.
#                                                      For TR parameters, see appropriate
#                                                      documentation in IP Configuration.

# The following are a set of default policyAction definitions based
#   on the precedence field of the TOS byte in the IP header
#   (refer to RFC 791 for detail).   For network domains that support
#   Differentiated Services (DS) definition (RFC 2474), the outgoing
#   TOS (aka DS code points) value needs to be updated.
policyAction   networkcontrol
{
     policyScope    DataTraffic
     OutgoingTOS    11100000      # Precedence bits (first 3 bits)
}

policyAction   internetwork      # encapsulated network control
{
     policyScope    DataTraffic
     OutgoingTOS    11000000      #
}

policyAction   crit-realtime     # realtime data
{
     policyScope    DataTraffic
     OutgoingTOS    10100000     #
}

policyAction   interactive1
{
     policyScope    DataTraffic
     OutgoingTOS    10000000
}

policyAction   interactive2
{
     policyScope    DataTraffic
     OutgoingTOS    01100000
}

policyAction   batch1
{
     policyScope    DataTraffic
     OutgoingTOS    01000000
}

policyAction   batch2
{
     policyScope    DataTraffic
     OutgoingTOS    00100000
}


# policyRule statement
#   This statement specifies characteristics of IP packets, that are used
#   to match to a corresponding policyAction.  In other words, it
#   defines a set of IP packets that should receive a particular service.
#   It also can be used to define an application (e.g. port number)
#   that has TR action to be enforced.
#
#   statement format:
#         policyRule                              s  # Policy rule name.
#         {
#            PolicyRulePriority                   i  # Priority of this policy rule.
#            ForLoadDistribution                  p  # Is the rule intended for load
#                                                    #  distribution?
#            SourceAddressRange               a1 a2  # Source IP address range.
#            DestinationAddressRange          a1 a2  # Destination IP address range.
#            SourcePortRange                  i1 i2  # Source port range.
#            DestinationPortRange             i1 i2  # Destination port range.
#            ProtocolNumberRange              i1 i2  # Transport protocol id range to which
#                                                    #  this policy rule applies.
#            ApplicationName                      s  # Local application/job name.
#            ApplicationData                      s  # Application data, used for Web QoS.
#            ApplicationPriority                  i  # Application specified priority.
#            ServerDomainName                     s  # HTTP request URL domain name.
#            UserNameId                           s  # User name requesting service.
#            UserQoSGroup                         s  # User group requesting service.
#            InboundInterface                   a|s  # Incoming interface for which
#                                                    #  this policy rule applies.
#            OutboundInterface                  a|s  # Outgoing interface for which
#                                                    #  this policy rule applies.
#            IncomingTOS                          b  # IncomingTOS value/mask.
#            ConditionTimeRange                   s  # Overall time range.
#            MonthOfYearMask                      b  # Months of year that this
#                                                    #  policy rule is active.
#            DayOfMonthMask                           b  # Days of month that this
#                                                        #  policy rule is active.
#            DayOfWeekMask                            b  # Days in a week that this
#                                                        #  policy rule is active.
#            TimeOfDayRange                           s  # Time of each day during which
#                                                        #  policy rule is active.
#            PolicyActionReference                    s  # Name of an action which this policy
#                                                        #  rule uses.
#         }
#   where:
#         s                                        (R):  Is the name of this policy rule (up to 32
#                                                         characters, truncated if longer).
#         PolicyRulePriority                       (O):  Priority of a rule, it determines the order
#                                                         of rule evaluation relative to others.
#                                                         Default is for Pagent to assign a priority
#                                                         based on rule's specificity.  The maximum
#                                                         value is 255.
#         ForLoadDistribution                      (O):  Set to TRUE if the rule is intended for a
#                                                         Sysplex Distributor (SD) distributing
#                                                         stack.  Use this for rules to be
#                                                         interpreted on the SD distributing
#                                                         stack.  Valid values are TRUE and FALSE.
#                                                         The default is FALSE.
#         SourceAddressRange                       (O):  From a1 to a2 where a2 >= a1, default is 0
#                                                         which is all inclusive.  a2 is optional.
#                                                         IPv4 or IPv6 addresses can be specified.
#         DestinationAddressRange                  (O):  From a1 to a2 where a2 >= a1, default is 0
#                                                         which is all inclusive.  a2 is optional.
#                                                         IPv4 or IPv6 addresses can be specified.
#         SourcePortRange                          (O):  From i1 to i2 where i2 >= i1, default is 0
#                                                         which is all inclusive.  The maximum value
#                                                         is 65535.  i2 is optional.
#         DestinationPortRange                     (O):  From i1 to i2 where i2 >= i1, default is 0
#                                                         which is all inclusive.  The maximum value
#                                                         is 65535.  i2 is optional.
#         ProtocolNumberRange                      (O):  From i1 to i2 where i2 >= i1, default is 0
#                                                         which is all inclusive.  The maximum value
#                                                         is 255.  i2 is optional.
#         ApplicationName                          (O):  Eee EZAPAGAT (pagent_at.conf) for detail
#                                                         description.
#         ApplicationData                          (O):  See EZAPAGAT (pagent_at.conf) for detail
#                                                         description.
#         ApplicationPriority                      (O):  See EZAPAGAT (pagent_at.conf) for detail
#                                                         description.
#         ServerDomainName                         (O):  See EZAPAGAT (pagent_at.conf) for detail
#                                                         description.
#         UserNameId                               (O):  See EZAPAGAT (pagent_at.conf) for detail
#                                                         description.
#         UserQoSGroup                             (O):  See EZAPAGAT (pagent_at.conf) for detail
#                                                         description.
#         InboundInterface                         (O):  Specifies a valid local interface.  Default
#                                                         is all inbound interfaces.  Either an IPv4
#                                                         address or an interface name can be specified.
#         OutboundInterface                        (O):  Specifies a valid local interface.  Default
#                                                         is all outbound interfaces.  Either an IPv4
#                                                         address or an interface name can be specified.
#                                                         NOTE: if both local inbound and outbound
#                                                         interfaces are specified, the corresponding
#                                                         rule won't be mapped to any traffic.  This
#                                                         is because S/390 implmentation of policy
#                                                         is as a server where traffic is destined to or
#                                                         originates from, not as a routing node where
#                                                      traffic enters one interface and departs on
#                                                      another.
#         IncomingTOS                           (O):  bbbbbbbb-n
#                                                      First 8 bits are incoming TOS value.  n is
#                                                      the number of significant bits in the TOS
#                                                      (1-8).
#         ConditionTimeRange                    (O):  yyyymmddhhmmss:yyyymmddhhmmss
#                                                      see description of ptpConditionTime in
#                                                      EZAPAGAT (pagent_at.conf) for details.
#                                                      Seconds are rounded to the nearest minute.
#                                                      Default is always.
#         MonthOfYearMask                       (O):  A mask of 12 0's and 1's representing
#                                                      January to December.  Default is all year.
#         DayOfMonthMask                        (O):  A mask of 31 0's and 1's, default is
#                                                      all month.
#         DayOfWeekMask                         (O):  A mask of 7 0's and 1's representing Sunday
#                                                      through Saturday.  For example, 0111110
#                                                      represents weekdays.  Default is all week.
#         TimeOfDayRange                        (O):  A series of time intervals (up to 25),
#                                                      separated by a comma.  Time starts at 0
#                                                      which is right after midnight.  Only hour
#                                                      and minute are allowed to be specified.
#                                                      Default is 24 hours.
#                                                      Example: TimeOfDayRange 0-8:30, 17:30-24
#                                                      means this policy is only active after
#                                                      midnight to 8:30AM, and from 5:30PM to
#                                                      midnight.
#         PolicyActionReference                 (R):  Example: policyActionReference interactive
#                                                      Up to 4 instances of this statement
#                                                      can be specified (one name per attribute) to
#                                                      associate this policy rule with different
#                                                      actions.  For instance, a policy rule
#                                                      can reference two actions, one for RSVP
#                                                      and one for DataTraffic (DiffServ).

# The following are a set of default policy rules defined
#   for different traffic types including Enterprise Extender (EE),
#   based on port numbers from the host side (390 server).
#   This set is not exhaustive but only represents the majority of traffic
#   in a typical network.
policyRule          routed           # ROUTED traffic
{
     protocolNumberRange 17
     SourcePortRange    520          # ROUTED port
     policyActionReference  networkcontrol
}

policyRule          ospf             # OSPF link advertisement traffic
{
     protocolNumberRange 89          # OSPF protocol number
     policyActionReference  networkcontrol
}

policyRule          tftpd           # TFTP traffic
{
     protocolNumberRange 17
     SourcePortRange    69          # TFTP port
     policyActionReference  batch1
}

policyRule          ftpd            # FTP traffic
{
     protocolNumberRange  6
     SourcePortRange    20 21       # Both FTP control and data ports
     policyActionReference  batch1
}

policyRule          telnetd         # telnet traffic
{
     protocolNumberRange  6
     SourcePortRange    23
     policyActionReference  interactive1
}

policyRule          web-httpd       # web traffic
{
     protocolNumberRange  6
     SourcePortRange    80
     policyActionReference  interactive2
}

policyRule          dns-udp         # domain name server udp traffic
{
     protocolNumberRange 17
     SourcePortRange    42
     policyActionReference  interactive1
}

policyRule          dns-tcp         # domain name server tcp traffic
{
     protocolNumberRange  6
     SourcePortRange    42
     policyActionReference  interactive1
}

policyRule          ntp             # NTP traffic
{
     protocolNumberRange  17
     SourcePortRange    123
     policyActionReference  crit-realtime
}

policyRule          EE-xid
{
     protocolNumberRange 17
     SourcePortRange    12000
     policyActionReference  internetwork
}

policyRule          EE-network
{
     protocolNumberRange 17
     SourcePortRange    12001
     policyActionReference  internetwork
}

policyRule          EE-highpri
{
     protocolNumberRange 17
     SourcePortRange    12002
     policyActionReference  interactive1
}

policyRule          EE-medpri
```

```
{
     protocolNumberRange 17
     SourcePortRange    12003
     policyActionReference  batch1
}

policyRule          EE-lowpri
{
     protocolNumberRange 17
     SourcePortRange    12004
     policyActionReference  batch2
}

# The following is a sample policy for enforcing Differentiated Services
#   parameters that describe a token bucket mechanism.  The action
#   establishes a token bucket traffic conditioner with a mean rate of
#   128 kilobits per second and a burst size of 4 kilobytes.  Any traffic
#   that exceeds these specifications will be sent as best effort, with
#   an accompanying TOS byte of '00000000'.  Note that this is just an
#   example: exact specifications depend on the characteristics of the
#   sending application and the underlying network.
#
# PolicyRule                               DiffServ_Rule1
#  {
#     DestinationAddressRange               211.40.100.0-211.40.100.255
#     SourcePortRange                       20-21
#     PolicyActionReference                 DiffServ_Action1
#     DayOfWeekMask                         0111110
#  }
#
# PolicyAction                             DiffServ_Action1
#  {
#     PolicyScope                           DataTraffic
#     OutgoingTOS                           01000000
#     DiffServInProfileRate                 128       # 128 Kbps
#     DiffServInProfileTokenBucket          64        # 64  Kbits
#     DiffServInProfilePeakRate             512       # 512 Kbits
#     DiffServInProfileMaxPacketSize        120       # 120 Kbits
#     DiffServOutProfileTransmittedTOSByte  00000000
#     DiffServExcessTrafficTreatment        BestEffort
#  }
```

A1.2 Sample Pagent TTLS Configuration

```
####################################################################
#  IBM Communications Server for z/OS                              #
#  SMP/E distribution path: /usr/lpp/tcpip/samples/IBM/EZAPAGFT    #
#                                                                  #
#                                                                  #
#  5694-A01 (C) Copyright IBM Corp. 2005.                          #
#  Licensed Materials - Property of IBM                            #
#                                                                  #
#  Status = CSV1R7                                                 #
#                                                                  #
#  /usr/lpp/tcpip/samples/pagent_TTLS.conf                         #
#                                                                  #
####################################################################


# Common Diagnostic Group that a problem Rule may use
# Shows AT-TLS events and result of each System SSL call
TTLSGroupAction grp_Diagnostic
{
 TTLSEnabled On
 Trace 32            # Log Error, Info, Event and Flow to syslogd
}


####################################################################
#                                                                  #
# Application using ephemeral ports                                #
#                                                                  #
####################################################################

TTTLSRule Generic_Outbound_Application
{
 LocalAddr                All
 RemoteAddr               All
 RemotePortGroupRef       Generic_Server_App_Ports
 Direction                Outbound
 TTLSGroupActionRef       grp_Diagnostic
 TTLSEnvironmentActionRef Generic_Client_App
}

TTLSEnvironmentAction Generic_Client_App
{
  HandshakeRole       Client
  TTLSKeyRingParms
  {
#   keyring           TOREXT
```

```
    keyring           TSLKEYRING
  }
  TTLSCipherParms
  {
    V3CipherSuites    TLS_RSA_WITH_3DES_EDE_CBC_SHA
  }
  TTLSEnvironmentAdvancedParmsRef Generic_Client_App_Adv_Prm
}
TTLSEnvironmentAdvancedParms   Generic_Client_App_Adv_Prm
{
    SSLV2             OFF
    SSLV3             ON
    TLSV1             ON
}

# This group defines the Application Server SSL ports
PortGroup  Generic_Server_App_Ports
{
 PortRange
 {
  Port    443
 }
 PortRange
 {
  Port   9443
 }
 PortRange
 {
  Port    1740
 }
}
############################################
TTLSRule LDAP_Client_Application
{
 RemoteAddr               All
 RemotePortGroupRef       LDAP_App_Ports
 Direction                Outbound
 TTLSGroupActionRef       grp_Diagnostic
 TTLSEnvironmentActionRef LDAP_Client_App
}

TTLSEnvironmentAction LDAP_Client_App
{
  HandshakeRole       Client
  TTLSKeyRingParms
  {
    keyring           LDAPKEYRING
  }
  TTLSCipherParms
  {
    V3CipherSuites    TLS_RSA_WITH_DES_CBC_SHA
  }
  TTLSEnvironmentAdvancedParmsRef LDAP_Client_App_Adv_Prm
}
TTLSEnvironmentAdvancedParms   LDAP_Client_App_Adv_Prm
{
    SSLV2             OFF
    SSLV3             ON
    TLSV1             ON
#   ClientAuthType    PassThru
#   ApplicationControlled ON
}

# This group defines the Application Server SSL ports
PortGroup  LDAP_App_Ports
{
 PortRange
 {
  Port    636
 }
}
```

A1.3 Sample Pagent Environment setup

```
PAGENT_CONFIG_FILE=/etc/pagent.conf
PAGENT_LOG_FILE=/tmp/pagent.log
```

A1.4 Sample Pagent Started Task JCL

```
//PAGENT   PROC
//*
//* IBM Communications Server for z/OS
//* SMP/E distribution name: EZAPAGSP
//*
//* 5694-A01 (C) Copyright IBM Corp. 1998, 2005
//* Licensed Materials - Property of IBM
//* "Restricted Materials of IBM"
//* Status = CSV1R7
//*
//PAGENT   EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//       PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//*
//* Example of passing parameters to the program (parameters must
//* extend to column 71 and be continued in column 16):
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c
/
//*            etc/pagent3.conf -l SYSLOGD'
//*
//* Provide environment variables to run with the desired
//* configuration.  As an example, the data set or file specified by
//* STDENV could contain:
//*
//*   PAGENT_CONFIG_FILE=/etc/pagent2.conf
//*   PAGENT_LOG_FILE=/tmp/pagent2.log
//*
//* For information on the above environment variables, refer to the
//* IP CONFIGURATION GUIDE.  Other environment variables can also be
//* specified via STDENV.
//*
//*STDENV   DD DUMMY
//* Sample MVS data set containing environment variables:
//*STDENV   DD DSN=TCPIP.PAGENT.ENV(PAGENT),DISP=SHR
//* Sample HFS file containing environment variables:
//STDENV   DD PATH='/etc/pagent.env',PATHOPTS=(ORDONLY)
//*
//* Output written to stdout and stderr goes to the data set or
//* file specified with SYSPRINT or SYSOUT, respectively.  But
//* normally, PAGENT doesn't write output to stdout or stderr.
//* Instead, output is written to the log file, which is specified
//* by the PAGENT_LOG_FILE environment variable, and defaults to
//* /tmp/pagent.log.  When the -d parameter is specified, however,
//* output is also written to stdout.
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```