

SAP XMII GETTING STARTED GUIDE

Applies to:

SAP xMII version 11.5

Summary

The following document is an excerpt from the SAP xMII help documentation which can be found online at http://help.sap.com/content/documentation/xapps/docu_xapps_mii.htm.

Author Bio

Joe Montagna who edited and collected these contents from various sources, was the Technical Services Manager for Lighthammer, running the training and support departments for the Lighthammer CMS product. After the SAP acquisition started working to integrate these departments into the SAP process. Presently Joe manages the SAP xMII (formerly Lighthammer) support team which is providing Primary and Development support of the xMII product.

Joe has 13 years experience in manufacturing and plant floor systems.

Table of Contents

Getting Started.....	4
SAP xMII Configuration Overview.....	4
SAP xMII System Configuration.....	6
SAP xMII Main Menu.....	6
Obtaining System Data Connection Status Information.....	7
SAP xMII Navigation Overview	7
Previous Version Name - SAP xMII 11.0 Name Mapping.....	9
SAP xMII Migration Frequently Asked Questions.....	11
Migration to SAP xMII 11.5.1.....	11
Migration to SAP xMII 11.5.....	11
Migration from Versions 10.x to Version 11.0	12
Installation Guide	14
SAP xMII Installation Guide.....	14
Step-by-step Installation.....	20
Applet Introduction	38
Applet Overview	38
Applet Object Model Reference	57
Quick Start Guide.....	61
Adding Database Connections.....	62
Creating Query and Display Templates	62
Using FrontPage to Generate an Output Page	64
Using the Page Generator.....	64
Template Editor	65
Template Editor General Reference.....	65
Data Request Guide.....	66
Query Editor General Reference.....	73
Tag Query Editor Reference	75
SQL Query Editor Reference	77
Alarm Query Editor Reference	78
XML Query Editor Reference	79

Aggregate Query Editor Reference	80
Xacute Query Editor Reference	81
OLAP Query Editor Reference	83
iChart Editor Reference	84
iSPCChart Editor Reference	85
iGrid Editor Reference	92
iBrowser Editor Reference.....	93
iTicker Editor Reference	93
HTML Development Add-In	94
Using FrontPage to Generate an Output Page	96
Script Assistant User Guide	97
Script Assistant.....	97
Standalone Wizards	101
Dynamic Page Generator.....	101
Related Content.....	102
Disclaimer and Liability Notice.....	102

Getting Started

SAP xMII Configuration Overview

The SAP xApp Manufacturing Integration and Intelligence (SAP xMII) offers a sophisticated yet easy-to-use software application for aggregating, transforming, and distributing plant and production information via XML and Web technologies. This document describes how to configure and utilize the SAP xMII server components, including the "Connectors" and "Builders" provided with the SAP xMII. Connectors provide the "input" from a wide range of data sources, and builders provide the "output" in a number of data formats. Additionally, a sophisticated set of display components, implemented as Java Applets, provide the "presentation" and user interaction layer for the SAP xMII. On top of the core system, other value-add optional components are available to further enhance and customize the SAP xMII solution.

The core of the SAP xMII is the SAP xMII "Servlet", which receives requests from client applications, utilizes the services of "connectors" to retrieve the data, and dispatches the output to the builders, which in turn send the resulting Web output back to the client application in the appropriate format (binary, XML, HTML, etc.). Java Servlets are server-side software components that are executed in coordination with requests/queries to the Web server.

This document will demonstrate how to build HTML pages and/or Web queries to use the connectors directly, however, most of the topics are relevant to the techniques by which the Graphic User Interface (GUI) applets included with SAP xMII interact with the various connectors. All of the properties/parameters exposed by the connectors and builders are identical to those utilized when configuring the user interface applets.

A connector can be thought of as the programmatic interface to a back-end data source such as a plant historian, an HMI or SCADA system, an I/O device, an MES or ERP system, or a relational database. The connectors included in your system will vary, but fall into one of the following categories:

Tag - Used to access historian, human-machine interface (HMI, SCADA, and I/O systems)

SQL - Used to access relational data in a wide range of database sources

Alarm - Used to access alarm/event management systems (HMI, building automation, network management, etc.)

XML - Used to access static or dynamic XML content (typical interface to ERP and related systems)

Aggregate - used to take data from multiple queries and combine them into a single document.

OLAP - used to query data from data warehouse, or cubed data, using MDX query syntax

SAP xMII Business Logic Services - used to run an SAP xMII Business Logic Services query and return the results to an SAP xMII visual component.

When you configure a server in the SAP xMII, you are setting up a specific instance of a connector. For example, a connector is the generic ability to access a relational database. Each actual database that you connect to is an SAP xMII server.

The SAP xMII connectors greatly simplify the mechanics of querying the data source, as there is a common set of parameters/properties for all connectors, and additional common behaviors for each "class" of connector as described above. The SAP xMII also provides a common set of capabilities for specifying date ranges, selecting output formats, and so on. One of the features of the SAP xMII that differentiates it from other solutions is the ability to perform "name space browsing".

In addition to providing mechanisms for reading data values, the connectors typically provide the ability to browse/query the underlying data source to indicate what capabilities it provides, what data

sources/tags/tables it can access, and so on (name space browsing). This capability, when accessed via the Template Editor in development mode or via ad-hoc query dialogs in the Java applets, provides a powerful means for quickly personalizing/customizing your SAP xMII implementation.

The connector, via the SAP xMII servlet, accepts parameters from the user, via Hypertext Transfer Protocol (HTTP) from a Web page hyperlink, a Web page form submittal, a Java applet, or other HTTP query source. It then creates a query from those parameters in a format that the underlying data source can understand. The connectors are highly optimized to cache database or system connections to provide maximum performance and scalability. A sophisticated connection pooling system is built into the SAP xMII core system.

By "tunneling" all data requests and responses inside of the HTTP protocol, firewall and network configuration headaches are eliminated. The SAP xMII is a multi-tiered application server, with presentation, business rules, and data access segmented into multiple subsystems, which can be scaled across multiple computers (even multiple platforms). In addition to the SAP xMII component-level security, the inherent security capabilities of the Web server can be used to protect Web pages, Web queries, administration pages, and other secure resources. Additionally, the SAP xMII allows you to author your Web page content.

The IDBC connector provides a way to query any standard SQL database using structured query language (SQL). The IDBC connector includes native JDBC drivers for Microsoft SQL Server (6.X and above), Sybase, and Oracle (7.X and above). The IDBC connector can also be configured to utilize other JDBC drivers, such as those provided by IBM for its DB2 and AS/400 databases, as well as many others. The SAP xMII also includes an ODBC bridge to link to other types of data sources for which an ODBC connection is the only option. The query capabilities are quite flexible, and can interface with complex SQL statements, database views, and even stored procedures in the target data system.

The SAP xMII "system connector" is provided with the system to allow the user to retrieve system level information and administer/manage the system remotely. There are several informational modes that provide different views of the system status including a real time look at what servers are currently active, usage statistics, error information, version information, and name space/configuration information such as a list of installed connectors and servers.

The end-user "front end" to the SAP xMII server system is most frequently implemented using the Java applets provided with the SAP xMII system. These applets provide a wide range of flexibility in both query creation and data display (refer to the Applet Reference for more details). The applets are also completely customizable and programmable, enabling very sophisticated user interfaces to be developed in minimal time. The real power of the SAP xMII is delivered in its breadth of functionality, ease of configuration and use, and in its ability to aggregate and present data from a wide range of data sources and on a large variety of target devices/platforms.

However, the SAP xMII is also quite capable of "on-the-fly" HTML, GIF, and XML content generation, as well as transferring data into CSV (comma-separated value) format. As an alternative to using the Java applets, hyperlinks can be built into a Web page or hand-held device application to provide server-generated, up-to-the-minute content on demand. By using standard HTML tags the user can set up queries to the SAP xMII Connector of choice and have the result displayed in the Web page. When combined with advanced HTML techniques such as server-side includes (SSI), "live" reports can be created that merge and integrate data from multiple data sources in multiple display formats.

Similarly, the back-end system can be queried directly using HTTP requests, providing the capability for external systems, such as Enterprise Application Integration (EAI) middleware, to query the SAP xMII for data, allowing plant-to-supply-chain integration via a single interface rather than dozens of customized and difficult to manage interfaces. The ability to output data in XML and other formats helps facilitate a platform-neutral data interchange mechanism. The SAP xMII offers dynamic server-side conversion of XML documents using Extensible Stylesheet Language Transformation (XSLT). This transformation capability enables the SAP xMII content to be transformed into any standard XML schema for interfacing to enterprise level applications. The SAP xMII can also initiate transactions to other systems through the use of the ActivePortal agent technology on a timed or event basis.

The SAP xMII is also capable of processing XML input via the XML connector, and feeding this data to other SAP xMII components such as the Java applets or the agent subsystem.

The SAP xMII utilizes XML internally to store system query and display templates, and other server-side persistent data. By utilizing an open format such as XML, systems integrators, OEM's and other application developers can automatically generate template information by adhering to the SAP xMII XML formats.

SAP xMII System Configuration

The product configuration is stored in SAP xApp Manufacturing Integration and Intelligence (SAP xMII) system files that are included with the installation. A complete set of editors, or pages, is supplied to administer and configure the system. This document explains the structure of the configuration and the basic layout of the editors.

There are a number of editors that perform specific tasks. Each editor has a corresponding service that controls access to that editor. For example, the Data Servers editor allows you to create, modify, and delete data sources. In order for a user to be allowed to use this editor, that user would need to be a member of a role that has been granted access to the corresponding service. By default, the Administrators role has access to all services. The access to specific services is configured in the System Security editor.

SAP xMII Main Menu

The main SAP xMII menu can be accessed via the following URL:

`http://<servername>/Lighthammer/Menu.jsp.`

Note: After the server name, the URL is case sensitive.

This main menu has links to all configuration options in the system. It is a dynamic menu, so only those administration functions to which a user has been granted access appear on the menu. Descriptions of the functional areas are listed below. A summary of the associated pages are available in the SAP xMII Navigation Overview which can be found in the xMII help documentation.

The system administration tools, like all configuration tools in SAP xMII, are Web-based and can be run on any Web browser. However, the target browser should support a minimum screen resolution of 1024x768 when using the various administration and template editing tools provided with the software. Additionally, the system font should be set to a small font size or the graphic elements will overlap due to inadequate room to fit all of the components on the page.

System Management

The System Management section includes system-wide configuration and monitoring tools.

Security Services

Security Services includes a link to the Security Manager system, as well as links to set up trusted servers and editors in order to grant access to data servers and system services. The Security Servers editor allows you to configure one or more Security Manager nodes if you choose to manage authentication services on a separate server or define backup security servers.

Data Services

Use the Data Services section to set up data source connections and create query templates to use against those data sources.

Visualization Services

Visualization Services contains a set of tools for creating display templates and other dashboard objects for use within the SAP xMII portal.

The Dynamic Page Generator and the Object Page Generator are productivity tools for generating HTML examples.

The Localization applet is used to define tokens when creating localized HTML pages.

You can use the Dynamic Graphics engine to create custom user interface widgets.

Wizard Install

Portal Services

The Portal Services section contains links to the Content Editor and the Navigation Editor. Both are used to define the SAP xMII navigation structure for users and roles.

Business Logic Services

Business Logic Services links to the xMII Logic Editor and the related Transaction Schedule Editor and Viewer.

MetricAlert Services

The MetricAlert Services section is a set of configuration windows that allow you to set up and subscribe to system alerts. See KPI and Alert Manager for more information.

Support

The Support section has links to the online help and the SAP xMII Technical Support Site.

[Obtaining System Data Connection Status Information](#)

If you click the Status button in the Data Servers Editor or select the Connection Status menu option from the Data Services menu, a new browser window listing the status for all of the currently enabled servers that required a connection pool appears. This tool provides a quick snapshot of which data sources are currently connected and functioning properly. If no connections are available for a given server, some type of configuration or network/communications problem is preventing SAP xMII from connecting to the server.

This display can be generated at any time by entering the following URL in a Web browser:

```
http://<server name>/Lighthammer/Illuminator?Service=SystemInfo&Mode=Status
```

SAP xMII Navigation Overview

The SAP xApp Manufacturing Integration and Intelligence (SAP xMII) main menu is the entry point to all configuration and content generation functions. The menu is divided into logical groups, designed around normal use of the administration and configuration environments. The SAP xMII menu is dynamic, and only those services for which you have been granted administration privileges are visible to you. The available menu options are listed below:

System Management

System Administration - configuration of server properties and behaviors

System Tasks- information on the default system tasks

Active Sessions - view of all current logged in users

License Viewer - view of server's license configuration

Java Runtime Status - view of the Java runtime statistics

Log Management

Log Configuration - configuration of General, Audit, and Signature logs

General Log - view of debug and error information

Audit Log - complete log of all configuration changes

Signature Log - log for support of electronic signature solutions

Proxy Log - log for support of proxy filtering

UDC Log - log for viewing universal data connector (UDC) servers information

Runner Log - log of xMII Logic Editor transactions

SOAP Runner Log - log for support of Web service calls to the server

Catalog Log - log of catalog service requests

User Log - log for support of the Event Logger action

System Stats Log - log of statistics for the system

Log Monitor - inline monitoring of log statements, similar to DBMON

Applet Debug - configuration screen for applet debug mode

Security Services

Security Manager - link to main menu for Security Manager

Trusted Servers - trusted server configuration

System Security - configuration for granting access to SAP xMII Services to roles

Data Access - configuration for granting access to SAP xMII data sources to roles

Security Servers - page to configure one or more Security Manager servers (for redundancy)

Data Services

Data Servers - configuration for data sources for SAP xMII application ("server" configuration)

SAP Server Configuration - configuration of SAP servers for use in the xMII Logic Editor

IDoc Listeners - configuration of IDoc listeners

Query Templates - query template editor

Time Intervals - time period and time shift editor

Time Period Schedules - time period groupings

Connection Status - view of current status of data sources with pooled connections

Visualization Services

Display Templates - display template editor

Dynamic Page Generator - simple to use applet creator/test tool

Object Page Generator - simple to use applet creator/test tool

Localization - editor for user defined localization strings for use in HTML pages

Dynamic Graphics - configuration environment for animating SVG objects for use in dashboards

Wizard Install

Portal Services

Content - name content editor

Navigation - editor for modifying home page navigation structures

Business Logic Services

Logic Editor - SAP xMII logic editor

Scheduler - view of scheduled jobs

Schedule Editor - set up for scheduled jobs

Metric/Alert Services

Mail Server - setup for mail server to handle e-mail alerts

Categories - category editor for metrics

Datasets - dataset editor for metrics; datasets are a way to use one query to feed multiple metrics and to reduce query load on back end systems

Metrics - specific metric definition editor

Monitors - editor for alert expressions

Scan Groups - editor for creating different frequency scan rates

System Alerts - view of alert log

Support

Help Documentation

SAP xMII Support Site

[Previous Version Name - SAP xMII 11.0 Name Mapping](#)

For those users who are looking for the same editors/pages found in previous versions, the following table shows what the previous menu names were versus the new SAP xMII menu names:

<i>Previous Name</i>	<i>New Location and Name</i>
Server Editor	Data Services Data Servers

System Administration Editor	System Management System Administration
Trusted Server Editor	Security Services Trusted Servers
Time Interval Editor	Data Services Time Intervals
License Editor	System Management License Viewer License updates are no longer made via a Web page but by updating the license file manually.
User Administration Editor	See the Security Manager documentation for more information.
Template Editor	The Template Editor has been broken into two separate editors to improve performance. Query templates are managed on the Template Editor page via Data Services Query Templates. Display templates are managed on the Template Editor page via Visualization Services Display Templates.
Dynamic Page Generator	Visualization Services Dynamic Page Generator
Object Page Generator	Visualization Services Object Page Generator
Localization Editor	Visualization Services Localization
Named Content Editor	Portal Services Content
Home Page Editor	Portal Services Navigation
Connection Status	Data Services Connection Status
Active Sessions	System Management Active Sessions

Login Page	See the Security Manager documentation for more information
Transaction Editor	xMII Logic Editor via Business Logic Services Transactions
Schedule Editor	Schedule Editor via Business Logic Services Scheduler
Animation Editor	Dynamic Graphics Editor via Visualization Services Dynamic Graphics

SAP xMII Migration Frequently Asked Questions

Migration to SAP xMII 11.5.1

Question: The Business Logic Services system properties configured in the Transaction Editor were removed. Where are these properties configured now?

Answer: You now configure proxy server settings on the System Administration page and Business Logic Services logging levels on the Log Configuration page. This change allows you to view logs and events for all components in a common logging infrastructure.

Question: The data types for some PDF action properties changed. What does this change affect?

Answer: When you open a transaction created in a version prior to SAP xMII 11.5, you must manually update the values for the following properties prior to executing the transaction:

Text Style and Alignment for the PDF Text action

Header Font Style and Text Font Style for the PDF Table action

Text Font Style and Alignment for the PDF Image action

Alignment and Barcode Height for the PDF Barcode action

Question: XPath links are evaluated differently in .NET and Java when the attribute or element repeats. In the .NET version, the last match was chosen. In the Java version, the first match is chosen. Why did this change?

Answer: The first match is more efficient and intuitive. If your existing transactions depended on the last match, those transactions may be affected.

Migration to SAP xMII 11.5

Question: Are the Read MSMQ Messages and Send MSMQ Message actions supported in 11.5?

Answer: Message queueing actions were removed from SAP xMII Business Logic Services. The Read MSMQ Messages and Send MSMQ Message actions are no longer supported.

Question: User-defined XSL transforms are no longer valid. What changed?

Answer: If you use the following reference to refer to built-in Java methods, you must change "com.lighthammer.Illuminator" to "com.lighthammer."

Change

```
<xsl:value-of  
select="java:com.lighthammer.Illuminator.common.LHUtilFunctions.parseBooleanValue(string($ShowTabs))"/>
```

to

```
<xsl:value-of  
select="java:com.lighthammer.common.LHUtilFunctions.parseBooleanValue(string($ShowTabs))"/>
```

Question: The Business Logic Services Scheduler Service settings changed. Why?

Answer: If a previous version of Business Logic Services (formerly Xacute) and the Business Logic Services scheduler is installed on your machine when you upgrade to SAP xMII 11.5, the Business Logic Services Scheduler Service is automatically reset to run manually. If you then uninstall 11.5, the old scheduler service is removed; however, it does not uninstall the scheduled transactions.

If you stop the 11.5 install and return to a previous version, run the Xacute Install Repair option to reestablish the Business Logic Services Scheduler Service.

Question: GIF and BMP image types were removed from 11.5. How does this change affect existing transactions with these image types?

Answer: GIF or BMP image types in existing transactions default to an image type of PNG in version 11.5 so they still work on execution. If the action is configured, the image type dialog defaults to "image/png."

Question: A Password Manager prompt appears on the Data Servers page when using Mozilla Firefox to save data server configuration changes. What is the correct response?

Answer: If you are using Firefox and a Password Manager prompt appears on the Data Servers page, select the No... or Never... option on the prompt. Your data server changes will be saved correctly.

If you inadvertently selected the Yes option on the prompt, you must clear your cache in order to properly save your changes.

Question: Will custom XSL stylesheets that use embedded Java calls to the LHUtilFunctions class package work in 11.5?

These files must be modified in order for them to work in 11.5.

For example:

```
java:com.lighthammer.Illuminator.common.LHUtilFunctions.parseBooleanValue
```

must be changed to

```
java:com.lighthammer.common.LHUtilFunctions.parseBooleanValue
```

[Migration from Versions 10.x to Version 11.0](#)

Question: I have been using the Guest account for some logins. How does it work in 11.0?

Answer: To automatically access a page, use a URL similar to the following:

```
http://<servername>/Lighthammer/Login.jsp?IllumLoginName=Guest&IllumLoginPassword=Guest&session=true&target=/Test/report.irpt
```

You need to set up a user in the Security Manager (using any account name) and assign the user to appropriate roles. Then under Guest Configuration in Security Manager, enter the name of the account you created. That account will then be treated as the guest account. When you log in as shown above, that mapped account and its role memberships are used. The target= is the relative URL that you wish to access. The session=true keeps the session, so only the first URL needs the login information. See the Guest Account topic for more information.

For information on logging out of the system without closing the browser, see Logging Out of SAP xMII in the help documentation.

Question: Is the SPC rules interpretation different in 11.0?

Answer: The default settings for the rule violations in 11.0 are based on the Nelson Rules. The interpretation of the rule violations has changed slightly.

For example, the default in 10.x for the Trend rule was 6 of 6 which required 7 points and 6 transitions in the same direction. After further investigation, 11.0 was changed to be consistent with industry standard interpretations of the Nelson Rules. In version 11.0, 6 of 6 means 6 points in the data set with 5 transitions in the same direction. You can change this setting by changing the configuration of the individual rule in the display. Changing the setting for the Trend rule to 7 of 7 will make the rule violations the same in both versions.

For further explanation, see <http://www.qualityamerica.com/knowledgecente/articles/CQEIVH3e.html>.

In Figure IV.27, Test #3 that shows 6 points and the 6th point is the alarm, not the 6th transition point (meaning the 7th point in the subset).

For more information on SPC rules, refer to the Statistical Process Control (SPC) Help file.

Question: My bookmarks to the system editors and administration tools no longer work. Where did these applications go?

Answer: The menu structure changed in 11.0. The main menu for developers and administrators can now be accessed using the URL <http://<server>/Lighthammer/Menu.jsp>. See the SAP xMII Navigation Overview for more information.

Question: How do I deploy third-party drivers (such as a third-party JDBC driver) in 11.0?

Answer: The deployment architecture of SAP xMII has changed, and drivers need to be deployed into different locations. Once the driver file is deployed, it is automatically found by the application, and no registration entries are required. Simply put the .jar or .zip file into the lib directory of the SAP xMII Web application. The directory is located under the Servlet Exec AS installation folder: ServletExec AS\se-[instance name]\webapps\default\Lighthammer\WEB-INF\lib.

Servlet Exec is installed under the Inetpub directory you entered during installation. The default location is: <Drive Letter>:\inetpub\ServletExec AS\webapps\default\Lighthammer\WEB-INF\lib.

To add a third-party driver to the Security Manager application, the process is similar but the Web application directory is named "LHSecurity" instead of "Lighthammer."

Typically, any new drivers require a restart of Servlet Exec in order to be recognized.

Note: If you are upgrading, the system backs up existing third-party JDBC drivers to the Lighthammer\lib_backup_[datetime] directory.

Question: I customized my document type definition (DTD)/default templates and now the changes I made are not used. Why?

Answer: The DTD files for the templates have been removed. The default settings are found in the default template files under the Templates directory: <Drive Letter>:\Lighthammer\Illuminator\Templates\Defaults.

These files that are delivered with the installation are considered system files and overwrite the existing files. However, your existing files are backed up and can be found in the previous installation backup location at <Drive Letter>:\Illuminator_backup_<datetime stamp>\Templates\Defaults. You cannot copy over your changes; this causes the applets to fail because of changes made to the default templates. You should not change the default applets, but you can create your own set of defaults that are used when you create new templates. If you change the default applets, your changes may be overwritten during upgrades.

Question: How do I automatically log in through a hyperlink?

Answer: The automatic login URL has changed. See Programmatic Logins in the xMII help documentation.

Question: What happened to demo mode?

Answer: The software no longer supports a one-hour demo mode. If you want a temporary license, you can get one from the self-service area of the support site at <http://support.lighthammer.com/>. You need your support site login information. On the support site, you can see your license and generate a temporary license file at any time.

Question: I am being automatically redirected to a login page when I attempt to access certain SAP xMII content. What changed in version 11.0?

Answer: The security filter in SAP xMII is used to authenticate and authorize users before they are allowed to access any content. The behavior in 11.0 is different but flexible. Without logging in to SAP xMII, you can define what content cannot be accessed by configuring the filter. For details on the security filter, see the Application Security chapter in the table of contents of the xMII help documentation.

Question: Prior to 11.0 debug messages and other Illuminator log messages could be found in the ServletExec logs. Where are these messages written in 11.0?

Answer: With the installation of 11.0, a <Drive Letter>:\Lighthammer\Logs directory is created. The directory includes an Illuminator.log file containing log records for the Illuminator application. Various migration, security, audit, and login log files are also included in the directory. To read more about SAP xMII 11.0 logging, refer to the help topics related to log configuration and viewing under the System Management chapter in the table of contents.

If SAP xMII does not appear to run or initialize, the ServletExec and ServletExecNative logs may provide information as to why.

Installation Guide

SAP xMII Installation Guide

Introduction

One of the key benefits of the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) is its ultra-thin client architecture that requires zero client administration. SAP xMII clients require a standard Web browser only which minimizes installation and management. No SAP xMII software needs to be installed on any of the client systems. Setup and installation of SAP xMII is performed entirely on your Web server. The client browser does require that a Java Runtime Environment (JRE) plug-in to be installed in order to use the full set of user interface objects. The supported JRE for a client machine is version 1.4.2 from Sun Microsystems, Inc.® (the latest release of 1.4.2 is recommended for the client).

Note: It is not recommended that the server application be installed on a server with a JSDK/JRE version greater than 1.4.2_07.

This SAP xMII installation is optimized to support Microsoft® Internet Information Services (IIS) on Microsoft Windows® 2003; however, Microsoft Windows® 2000 Server may also be used. Support for any non-Windows platforms and/or support for any servlet engine other than that supplied with the product requires special licensing arrangements. There is no end user support for these other platforms.

SPC data analysis requires a database to be installed and configured. The recommended configuration is to install Microsoft SQL Server directly on the Web server.

Note: You do not need a database if you are not using SPC data analysis. The core SAP xMII functionality stores its configuration data in operating system files.

The SPC data analysis option benefits from a high performance CPU.

SAP xMII includes an add-in for Microsoft FrontPage® and one for Macromedia® Dreamweaver® that enables the use of these tools as a unified development environment for your SAP xMII application. Refer to the Installing the FrontPage Add-In section of this topic for more information. If you are using Macromedia® Dreamweaver®, refer to the Installing the Dreamweaver Add-In section of this topic for more information.

System Requirements

Prior to installing SAP xMII, you must ensure that your server meets the following minimum requirements:

	<i>For 1-10 Concurrent Users</i>	<i>For 11-100 Concurrent Users</i>	<i>For 101-500 Concurrent Users</i>	<i>For 501+ Concurrent Users</i>
Processor	1.8 GHz or faster	dual processor recommended; 1.8 GHz or faster	dual processor, 1.8 GHz or faster	dual processor, 1.8 GHz or faster
RAM	See Web Server Documentation (512 MB minimum recommended)	See Web Server Documentation (1024 MB minimum recommended)	See Web Server Documentation (4096 MBytes minimum recommended)	See Web Server Documentation (8192 MBytes minimum recommended)
Operating System	Windows 2003 Server -or- Windows 2000 Professional with Service Pack 4 or greater	Windows 2003 Server -or- Windows 2000 Server with Service Pack 4 or greater	Windows 2003 Server -or- Windows 2000 Server with Service Pack 4 or greater	Windows 2003 Server -or- Windows 2000 Server with Service Pack 4 or greater
Web Server Software	IIS 6.x or 5.x	IIS 6.x or 5.x	IIS 6.x or 5.x	IIS 6.x or 5.x
Java Virtual Machine	Sun JSDK 1.4.2_07	Sun JSDK 1.4.2_07	Sun JSDK 1.4.2_07	Sun JSDK 1.4.2_07
File System	NTFS	NTFS	NTFS	NTFS

The required system resources vary based on your applications; however, for sizing purposes, an estimated 2-4 MB per simultaneously connected user can be used.

Virtually any Web browser, including Microsoft Internet Explorer, Netscape® Navigator, Netscape® Communicator, Opera®, and Mozilla® Firefox®, can be used with SAP xMII. We recommend that you use the most recent versions of these products. As stated above, the client should have the Sun JRE plug-in installed to the browser.

The Sun® Java Runtime Environment (JRE) must be installed on the server prior to the SAP xMII installation (It is recommended that you download and install the full JSDK, instructions can be found below).

Note: Turn off the Java automatic update mechanism via the Java Plug-in Control Panel.

It is strongly recommended, although not required by SAP xMII, that the local volume where you are installing SAP xMII be formatted with the NTFS file system. This will enable a high degree of security control over your Web page content via direct integration with the Windows NT or Windows 2000 and IIS security and authentication subsystems.

If you will be using the SAP xMII FrontPage add-in, you must be running Microsoft FrontPage 2000 and must have the FrontPage 2000 server extensions installed on your Web server. Refer to the FrontPage section for detailed instructions on how to install these products.

System Installation Prerequisites

You will need to obtain a license file in order to run SAP xMII. You may request a license file at <http://support.lighthammer.com>, or through support@lighthammer.com. See the license manager section of the help for more information.

The Sun JSDK 1.4.2_07 must also be downloaded and installed on the server prior to installing SAP xMII. You can download the JSDK directly from http://java.sun.com/products/archive/j2se/1.4.2_07/index.html. Press on the "Download J2SDK" link on this page, save it to disk, and install the JSDK. A link to this site is also available on the xMIIInstall.htm on the root directory of the installation media (see below step by step instructions for more information).

In IIS, you will need a virtual scripts directory created. From within Internet Information Services (IIS) Manager under Administrative Tools, go to your web site where you wish to install SAP xMII (typically the Default Web Site). If the virtual directory does not exist, create it directly under the InetPub directory where IIS is installed (by default, this would be C:\inetpub\Scripts). It should have Read permissions, and Execute Permissions of Scripts and Executables. To check this, right click on the directory, and select Properties.

Windows NT

You must install a Web server prior to installing SAP xMII. Microsoft Web Server 4.0 is available as a free add-on to Windows NT Server or Workstation as part of the Windows NT 4.0 Option Pack. You can download the option pack or get it on CD-ROM from your local Microsoft reseller. If you are installing SAP xMII on an Windows NT Server system, you will install IIS 4.0. If you are installing SAP xMII on a Windows NT Workstation machine, you will install Personal Web Server (PWS) 4.0.

Detailed information on installing IIS is provided with the Windows NT Option Pack. During the NT Option Pack installation it is important to select the Custom installation option and verify that the following components are selected for install:

Internet Information Server (IIS)

Internet Service Manager

If you will be using ODBC data sources, it is suggested that you obtain the latest Microsoft Data Access Components from the Microsoft Data Access Web site and install them on your Web server. Note that these

are not required to access Microsoft SQL Server or Oracle, as SAP xMII will install direct, high-performance drivers for these databases.

If you are running Windows NT 4.0 and Internet Explorer 4.x on your Web server, it is recommended that you update certain system files before installing SAP xMII. It is advised that you run a Windows Update and install all recommended updates before proceeding.

Note: Installation of IIS may cause the removal or overwrite of some of the Windows NT Service Pack components. It is recommended that you reinstall Service Pack 6a or greater on NT systems, prior to proceeding to the installation of the SAP xMII application.

Windows 2000

If you are installing SAP xMII on an MS Windows 2000 Server system, you will install IIS 5.0. If you are installing SAP xMII on an MS Windows 2000 Professional system, then you will install Personal Web Server (PWS) version 5.0. Microsoft Windows 2000 will typically install the IIS 5.0 components you will need to run SAP xMII.

The Microsoft Web servers for Windows 2000 systems are available as part of the installation process of the operating system. If you already have the operating system installed, without the Web server, you may run the additional setup required by choosing "Add/Remove Windows Components" after starting the Add/Remove Programs service available within Control Panel (Internet Information Services (IIS) is the required component).

If you will be using ODBC data sources, it is suggested that you obtain the latest Microsoft Data Access Components from the Microsoft Data Access Web site and install them on your Web server. Note that these are not required to access Microsoft SQL Server or Oracle, as SAP xMII will install direct, high-performance drivers for these databases.

Note: Installation of IIS may overwrite or cause the removal of some of the Windows 2000 service pack components. It is recommended that you reinstall Service Pack 4 or greater on Windows 2000 systems, prior to proceeding to the installation of the SAP xMII application.

Windows 2003

By default, Windows 2003 Server has certain locked capabilities that must be changed after SAP xMII installation in order for SAP xMII to run. You must create a virtual directory in IIS with the proper privileges so that the servlet engine will run. Complete the additional steps for windows 2003 section after the Step-by-step Installation.

Windows XP Professional

Microsoft Windows XP Professional is not a supported server platform for production environments. It should be used for demonstrations and limited evaluations only.

If you are installing SAP xMII on a Microsoft Windows XP Professional system, you will install IIS 5.1. IIS is not installed by default with the operating system. You will have to go to the Add/Remove Windows Components section and choose to install it. Select the World Wide Web Service and press Details, then check the Scripts Virtual Directory.

When installing IIS, you MUST do a custom install and choose to install the Scripts Virtual Directory. Without this virtual directory SAP xMII WILL NOT run. Additionally, if you have installed any Windows XP Service Packs, it will be necessary to reinstall them after adding IIS.

If you are using SQL Server 2000, you should have Windows XP service pack 2 and SQL Server 2000 service pack 3 installed.

Step-by-step Installation

For a new installation, see Installing SAP xMII.

For an upgrade from an ISAPI version, see Installing SAP xMII: Upgrading from an ISAPI Version.

For an upgrade from an AS version, see Installing SAP xMII: Upgrading from an AS Version.

If you do not know if you have an ISAPI or AS version installed, complete the following steps:

Go to Control Panel -> Administrative Tools -> Services.

If the ServletExec service appears in the list, an AS version is installed. Otherwise, an ISAPI version is installed.

Additional Installation Steps for Windows 2003

Complete the following steps to allow the servlet engine to run as a recognized Web service extension.

Choose the Web Service Extensions folder.

Click Add a new Web service extension.
The New Web Service Extension window appears.

Enter ServletExec as the extension name, and click Add.
The Add file window appears.

Browse to the C:\inetpub\Scripts location and select ServletExec_Adapter.dll as the required file.

Select the Set extension status to Allowed check box.

Click the OK button.

Installing the FrontPage Add-in

The SAP xMII add-in for Microsoft FrontPage 2000 and Microsoft FrontPage 2002 is installed in the \Lighthammer\Illuminator\Wizards directory on the drive on which you installed the SAP xMII application.

In order to use the add-in, it must be installed on each PC that will be running FrontPage. On computers other than the Web server, a Web package installation is available on the SAP xMII Web server: <http://<servername>/Lighthammer/WizardInstall/IlluminatorWizard.exe>. You can save the file locally and run it, or you can select Open and run it directly from the Web server.

Note: Windows 2003 Server requires the FrontPage 2002 Server extensions.

Once the file is installed (the default is <SysDrive>:\Lighthammer\Illuminator\Wizards), open FrontPage. Select the /Tools/Add-Ins menu option. If you already had a previous version of the Add-in installed (prior to Illuminator 9.5) highlight the SAP xMII Wizard selection and press the Remove button. If this is a new install, skip to the enable step below.

To enable the new add-in, click the Add button, and browse to the location where the Illum10FPWizard.dll file resides (the default location is <SysDrive>:\Lighthammer\Illuminator\Wizards).

The add-in is automatically loaded each time FrontPage is started. The add-in includes the SAP xMII Object, the Script Assistant, and links to the SAP xMII Template Editor and the System Administration Editor.

Installing the Dreamweaver Add-in

The SAP xMII add-in for Macromedia Dreamweaver is installed in the \Lighthammer\Illuminator\Wizards directory on the drive on which you installed the SAP xMII products.

NOTE: In order to use the add-in, this file must be installed on each PC that will be running Dreamweaver. On computers other than the Web server, a Web package installation is available from the SAP xMII Web server: <http://<servername>/Lighthammer/WizardInstall/IlluminatorWizard.exe>

Use the following steps to enable the Dreamweaver add-in:

Copy the C:\Lighthammer\Illuminator\Wizards\IllumDWWizard.dll file to your local installation of the Dreamweaver MX program, JSExtensions directory (the default is C:\Program Files\Macromedia\Dreamweaver MX\Configuration\JSExtensions).

Copy the C:\Lighthammer\Illuminator\Wizards\Illuminator Object.htm file to C:\Program Files\Macromedia\Dreamweaver MX\Configuration\Commands.

Copy the C:\Lighthammer\Illuminator\Wizards\Illuminator Script Assistant.htm file to C:\Program Files\Macromedia\Dreamweaver MX\Configuration\Commands.

You should then be able to go into Dreamweaver and find the links at the bottom of the Command menu. If selecting SAP xMII Object does not display the plug-in, open a command prompt and type "cd\Lighthammer\Illuminator\Wizards" to navigate to the folder where the DLLs are located. Then type "regsvr32 IllumDWWizardGUI.dll" which should register the dll. Dreamweaver should then prompt you for the SAP xMII server name or internet protocol (IP) address (it defaults to localhost) when calling up the plug-in; this is similar to the standalone version of the plug-in.

You can then configure Dreamweaver for editing SAP xMII Report pages (.irpt).

Installing the iText Library for PDF Processing

SAP xMII supports the generation of PDF documents for both the documentation of xMII Business Logic Services transactions and user-defined output reports. The third-party library used for PDF processing is named iText. Version 1.2.3 has been tested with the system; therefore, it is the recommended version. For more information on iText, see www.lowagie.com/iText. The PDFActions.jar and component XML file are included in the SAP xMII build. However, in order to use PDF processing, you must complete the following steps:

Download the iText library version 1.2.3 via <http://prdownloads.sourceforge.net/itext/itext-1.2.3.jar?download>.

The file must be placed in two locations. Rename the file iText.jar and save it to ServletExec AS/se-xMII/webapps/default/Lighthammer/CMSLogicEditor and ServletExec AS/se-xMII/webapps/default/Lighthammer/WEB-INF/lib. The PDFActions.jar should already exist in both locations.

Note: The iText.jar file is case sensitive. It must match the entry in the XML file.

Move the PDFComponentCatalog.xml file from Lighthammer/Xacute/Components/optional to Lighthammer/Xacute/Components.

Restart the Servlet Exec AS service.

The PDF Actions option will then appear in the xMII Logic Editor navigation menu, and the File menu includes the Generate Documentation option for the transactions.

Initialization

By default, SAP xMII starts up (initializes) when the NT Service starts up. There is a time delay for the initialization in order for various system services to start up as before SAP xMII initializes. It is important that the timing is correct. The default delay is 30 seconds. This can easily be changed if it is causing problems.

A symptom of this problem is a class not found error in the Servlet Exec logs, because not all systems were initialized before SAP xMII was started, causing some paths to be incorrect.

Until the delay time is complete, it is recommended that you do not attempt to query SAP xMII. This includes accessing the SAP xMII main menu. Keep in mind that the delay is in addition to the delay caused by the NT

Service start up time. See the Server Logging section to see how to access the Servlet Exec Logs. See the section on SAP xMII Initialization in the system configuration guide for more information on this topic.

Time Synchronization of Servers and Clients

In order for SAP xMII to operate correctly, the system clocks on the Web server, data servers, and client browsers should be maintained in reasonably close synchronization.

In the case where other applications on the Web server prevent the enabling of automatic daylight savings time adjustment, you must configure the time zone to use the option that does not include the auto adjust checkbox [e.g., use the (GMT-05:00) Bogota, Lima, Quito time zone instead of the (GMT-05:00) Eastern Time (US & Canada) time zone].

Server Port Numbers

If SAP xMII is run on a port other than port 80, add the port number to the server Start menu shortcuts located in the Webapps\Default\Lighthammer and Webapps\Default\LHSecurity directories.

Installation Log Files

There are three installation logs created when installing SAP xMII. If you experience problems during your installation or have questions about installation locations, you can review these log files.

SAP_xMII_Install_date_time.log is an overall general log file for the installation located in the Lighthammer\Logs directory.

CMSServicesInstall_date_time.log contains details relating to the installation of your SAP xMII services and is located in the Lighthammer\Logs directory.

xMIISecurityInstall_date_time.log contains details relating to the installation of Security Manager and is located in the Lighthammer\Logs directory.

Step-by-step Installation

Installing SAP xMII

Note: If you are upgrading from an AS version, see Installing SAP xMII: Upgrading from an AS Version. If you are upgrading from an ISAPI version, see Installing SAP xMII: Upgrading from an ISAPI Version.

Note: If an error occurs during installation, review the install logs for more information.

Make sure you are logged on to the server machine using an Administrator account.

If you are running Windows NT, stop the Web server and IIS via the Control Panel Services applet.
OR

If you are running Windows 2000 or Windows 2003, stop the Web server and IIS via Start -> Programs -> Administrative Tools -> Services by stopping the IIS Admin Service.

This process automatically stops dependent services such as the World Wide Web Publishing service, FTP service, and SMTP service.

If you have CD media, insert the SAP xMII CD into your CD-ROM drive.

Auto-run opens the SAP xMII Installation page in your default browser.

OR

If you downloaded the software from the SAP Service Marketplace, extract the downloaded archive to a local directory and then open the xMIIInstall.htm page in the chosen directory.

Note: If you are using a CD and auto-run fails, you can open the xMIIInstall.htm page from the CD root. A link to the Sun Java 2 SDK installation is also available on this page. Once the JSDK is installed, you can run SETUP.EXE directly from the installation folder.

If you already installed the Sun JSDK 1.4.2_07 on the server, continue to step 5.
If a prior version of the Sun JDK or JRE is installed, uninstall the JDK/JRE and all references to Sun Java products including the following:

Java 2 Runtime Environment (JRE)

Java 2 Software Development Kit (SDK)

Residual folders and files under the root directory (i.e., C:\jdk1.3.1_01) or under \Program Files\Java

Registry entries (via Start | Run, enter Regedit in the Open field, select HKEY_LOCAL_MACHINE\SOFTWARE, and delete the JavaSoft folder and all its sub-directories)

Then install the Sun JSDK 1.4.2_07.

If the Sun JSDK 1.4.2_07 is not installed, download the JSDK from http://java.sun.com/products/archive/j2se/1.4.2_07/index.html. Click Download J2SDK, save the file to disk, and run the setup from the local disk.

Select the SAP xMII 11.5 installation link.
The InstallShield Wizard for SAP xMII screen appears.

If you did not stop the IIS services, a dialog appears. Click Yes to stop the service.
Another dialog appears.

Click OK to acknowledge that the IIS services were stopped successfully.

The Welcome screen for the InstallShield Wizard appears. Click Next.

Note: You may click Cancel at any time to stop the installation.

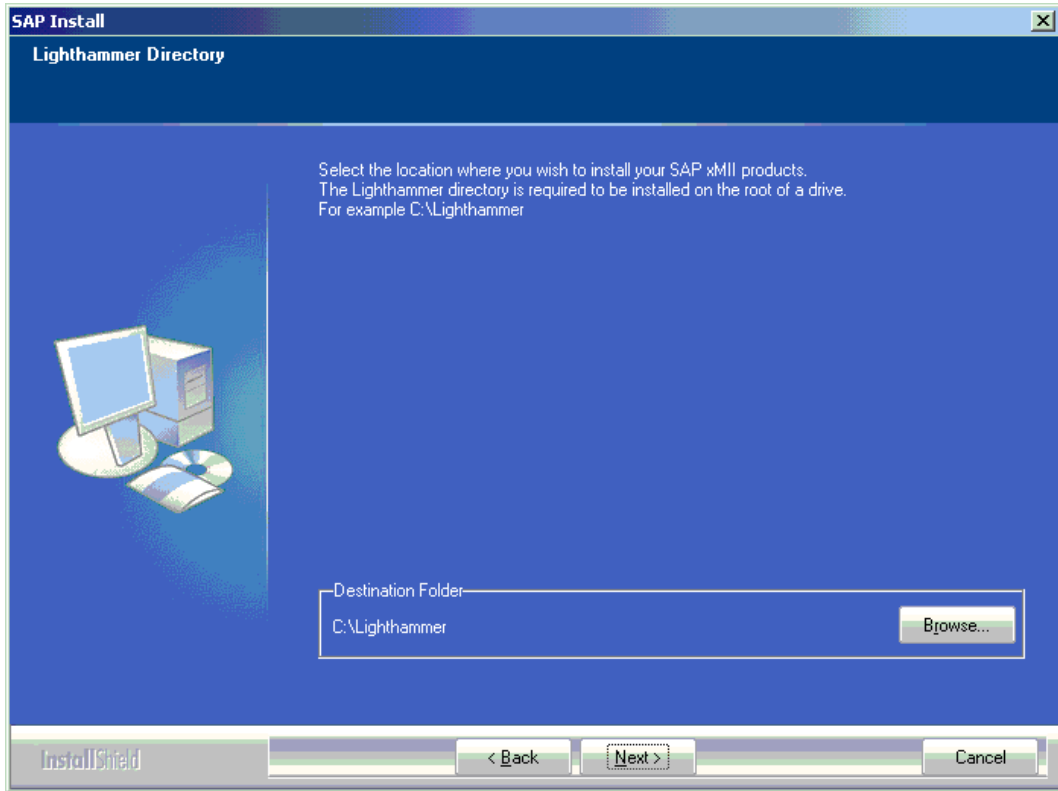
The License Agreement window appears. Read the license agreement carefully.
Click Yes to accept the agreement and continue the installation.

OR

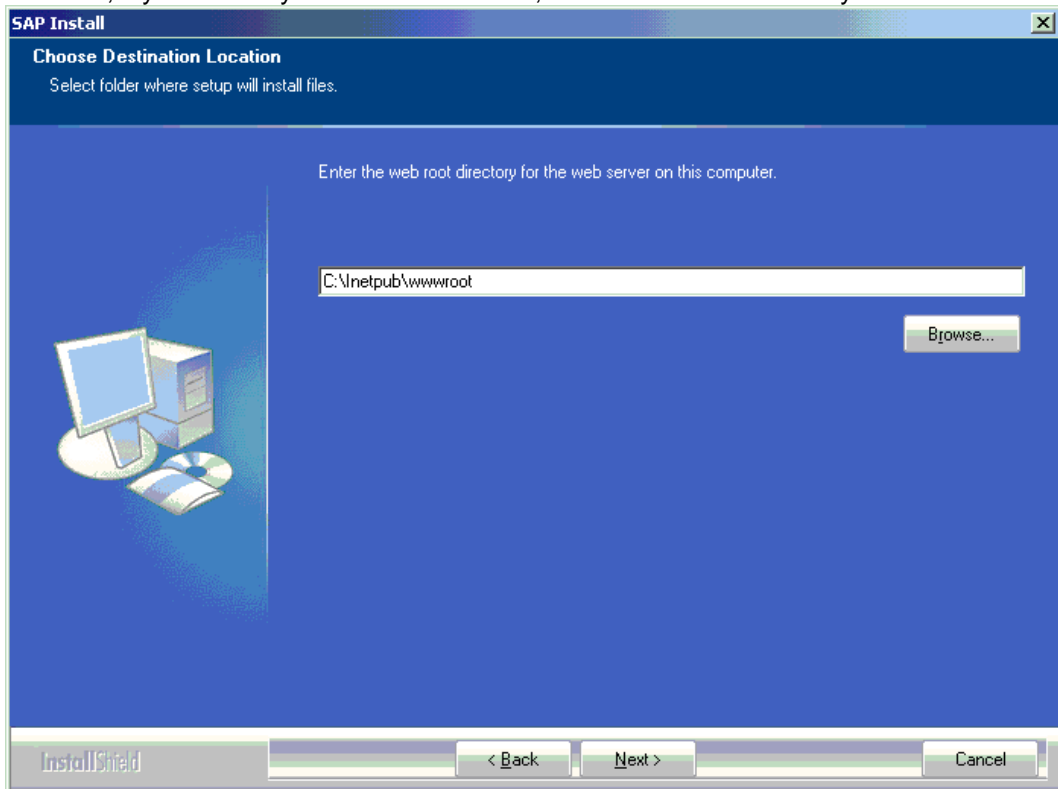
If you do not accept the license agreement, click Back. The setup stops and InstallShield Wizard closes.

The Lighthammer Directory screen appears. Choose the destination folder where you want to install the SAP xMII application. The default destination folder is acceptable, although it is recommended that you install the Lighthammer directory to the same drive on which you installed IIS. The Lighthammer directory should be installed on the root of the drive you choose. You should not install it in locations such as Program Files.

Click Next.



The Choose Destination Location screen appears. The default location of C:\inetpub\wwwroot is acceptable; however, if you moved your default Web root, enter the correct directory. Click Next.



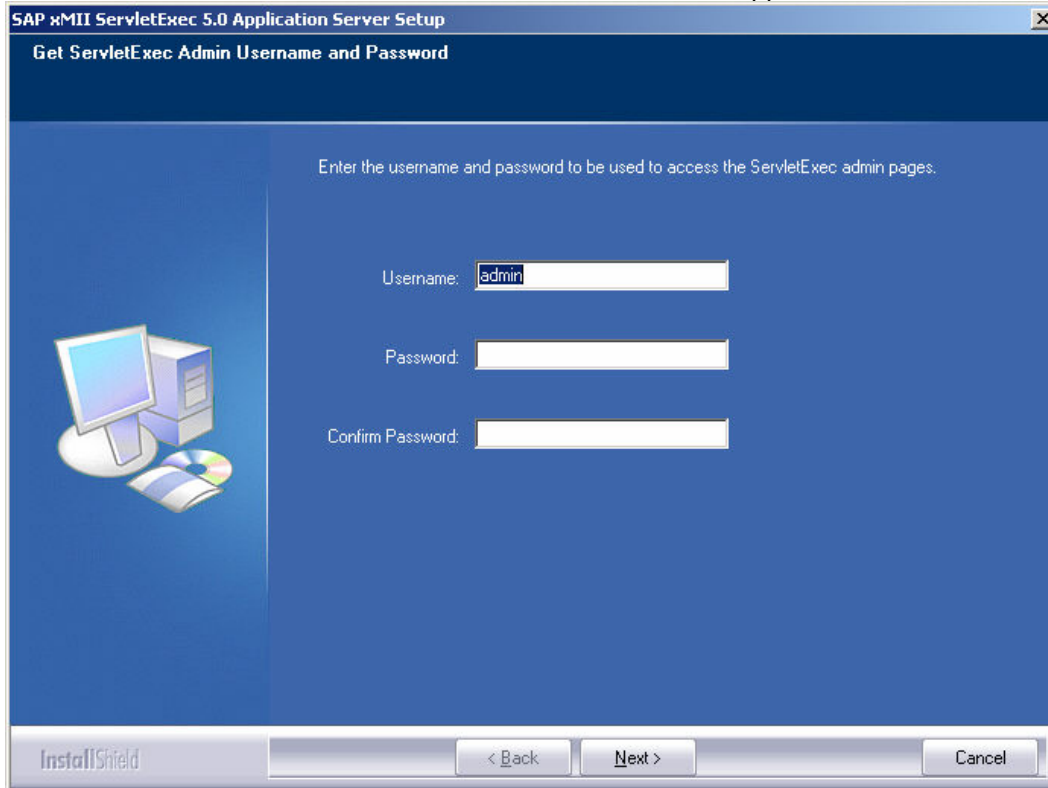
A screen appears. Click Next to set the destination folder as the default and initiate the ServletExec AS installation.

It may take several minutes. You may see one or more windows open and close during this process.

Wait for the ServletExec AS install to complete.

Click Yes to stop the IIS Admin Service.

The Get ServletExec Admin Username and Password screen appears. Enter the username and password.



SAP xMII ServletExec 5.0 Application Server Setup

Get ServletExec Admin Username and Password

Enter the username and password to be used to access the ServletExec admin pages.

Username:

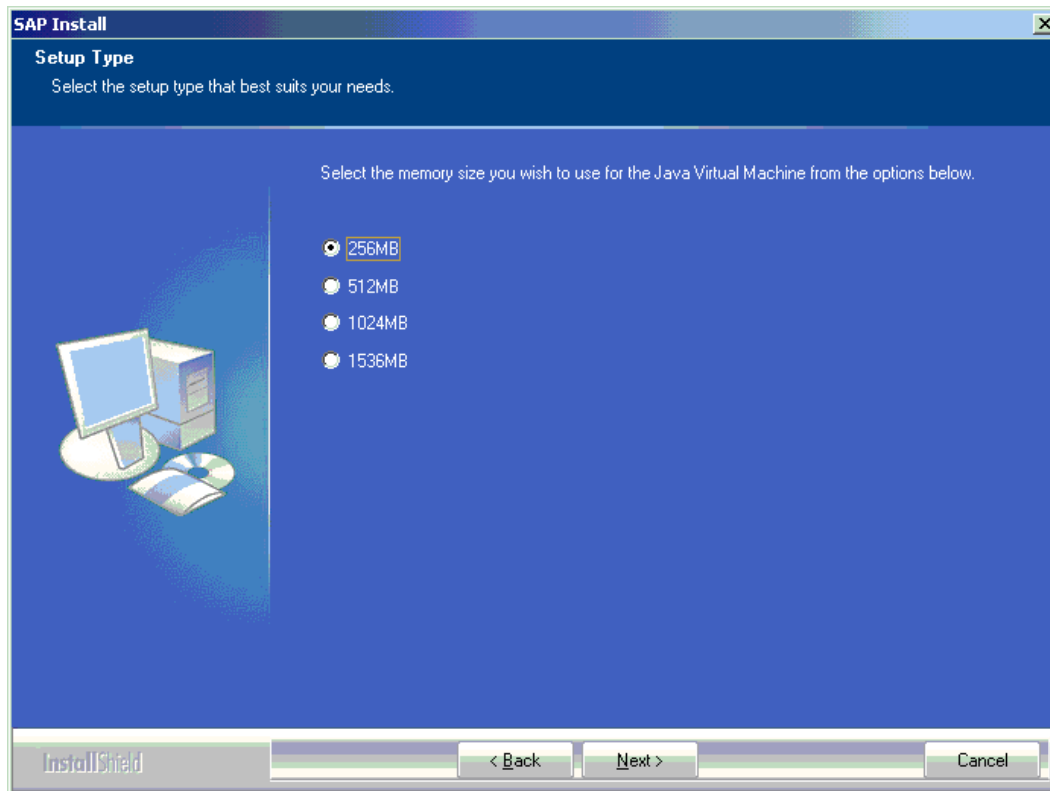
Password:

Confirm Password:

InstallShield

< Back Next > Cancel

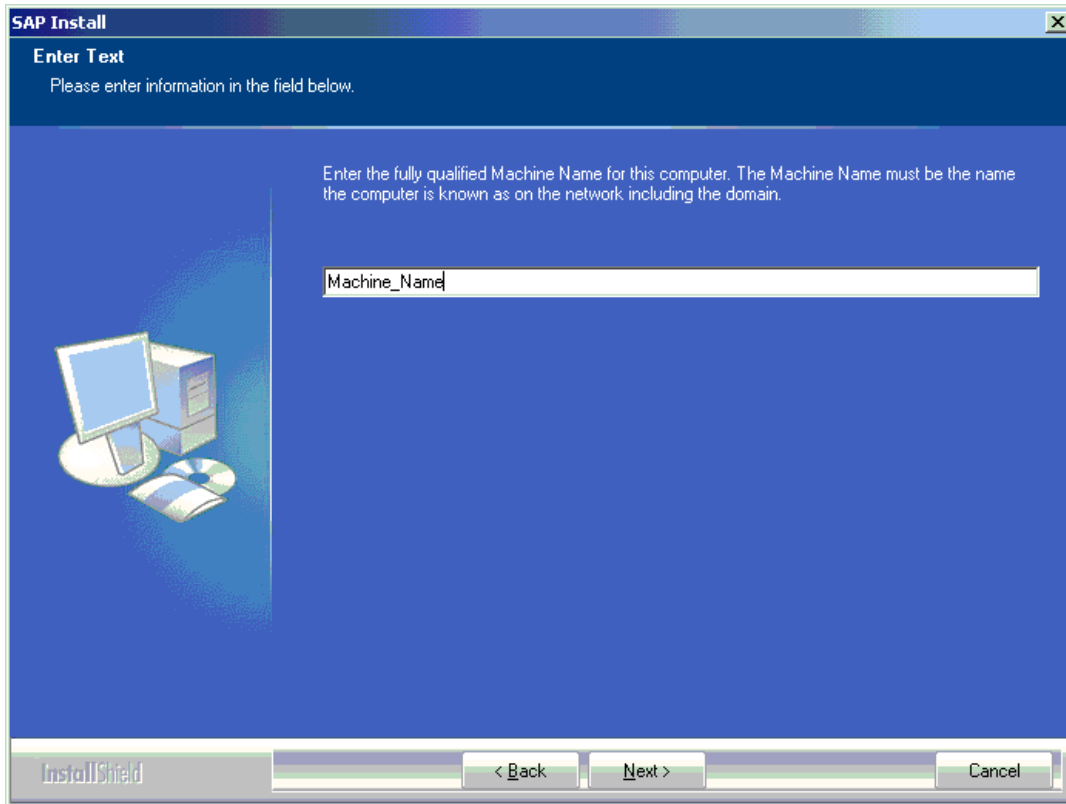
The Setup Type screen appears. Select the appropriate option button for the amount of RAM you want to designate to SAP xMII via ServletExec AS. It is suggested that a minimum of 512 MB be selected. If it is a dedicated SAP xMII server, we suggest one-half of the server's RAM as a guideline.



Note: Once the installation is complete and SAP xMII is operating properly on ServletExec Application Server, it is no longer necessary to restart IIS in order to restart SAP xMII. Go to the Services console, and restart the ServletExec-<InstanceName> (default name is ServletExec-LHCMS) service. This process reloads SAP xMII.

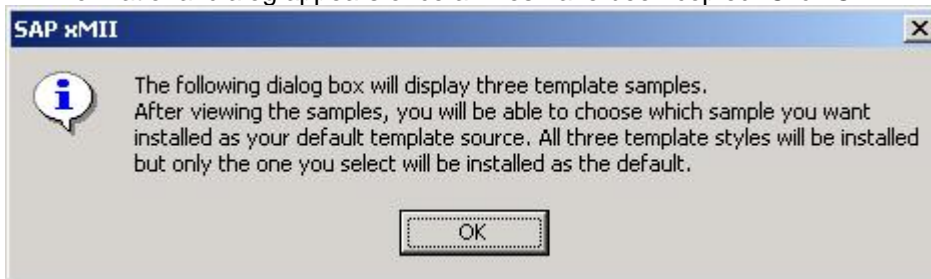
Enter the fully qualified machine name (for example, MACHINE_NAME.Domain.com) of the machine on which you are installing SAP xMII. The fully qualified machine name is required to identify the location of the security server in a multiple domain and/or third-party single sign-on scenario. If you are not implementing across domains or using third-party single sign-on, the machine name (without the domain name) is sufficient.

Note: If SAP xMII is run on a port other than port 80, you must add the port number to the machine name in the SAP Install window.

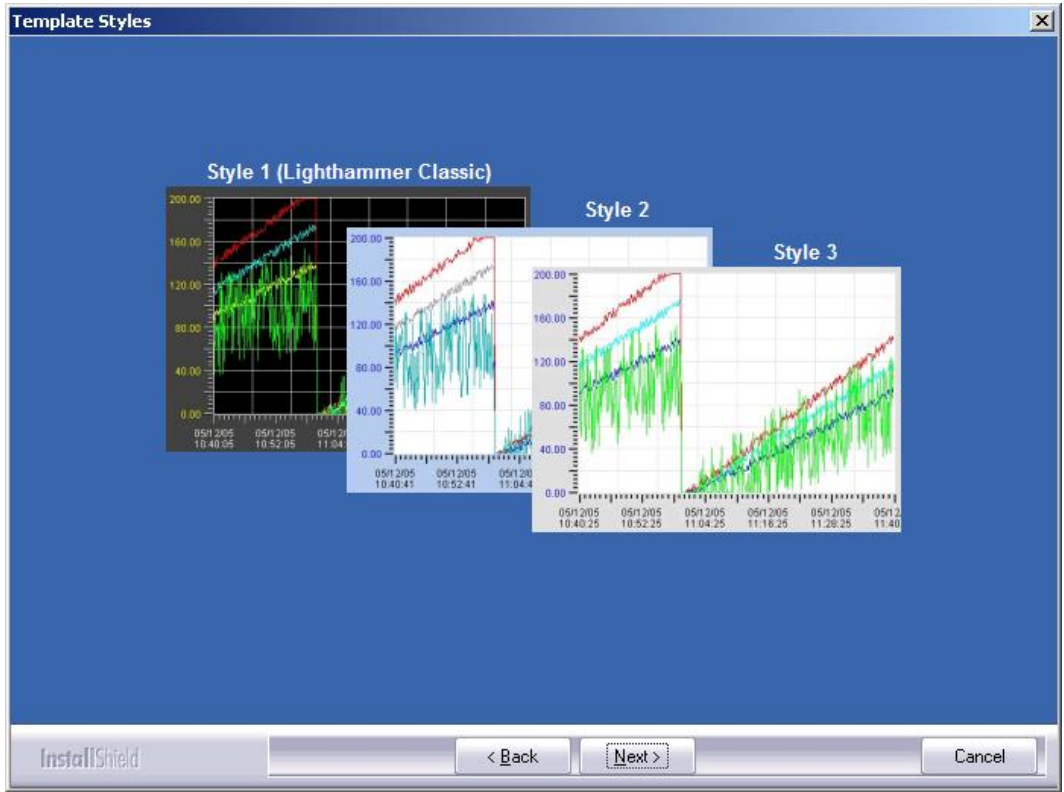


The installation begins copying files and installing. If an error occurs, the InstallShield stops and the setup does not continue. This process may take several minutes.

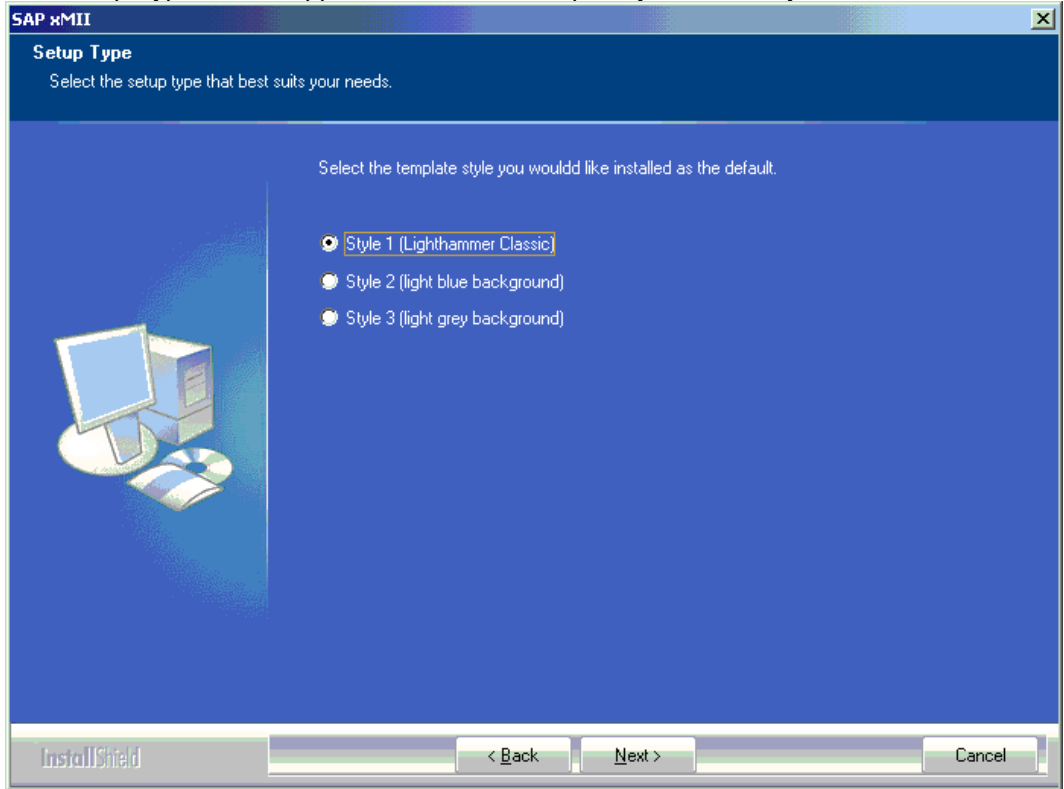
An informational dialog appears once all files have been copied. Click OK.



The style choices are displayed. All templates are installed on your machine, but only one is used as the default template style. Click Next to make your template selection.



The Setup Type screen appears. Select the template you want as your default.



After the installation is complete but before using SAP xMII, stop the IIS Admin Service and then restart the IIS Admin Service and the World Wide Web Publishing Service.

In Windows NT, open Windows Control Panel -> Services.

OR

In Windows 2000 and Windows 2003, choose Start -> Programs -> Administrative Tools -> Services.

Highlight IIS Admin, and click Stop to stop this service and all dependent services.

Restart the World Wide Web Publishing Service (and any other necessary Web services such as FTP or SMTP). The IIS Admin Service is automatically restarted. In Windows 2000 and Windows 2003, all Web services can be managed through Start -> Programs -> Administrative Tools -> Internet Services Manager.

Before accessing SAP xMII you also need to set up your license.

Copy your lhcms.lic license file to the C:\Lighthammer\License directory.

Restart your ServletExec-{InstanceName} (default name is ServletExec-LHCMS) NT service in the Windows Services screen.

Note: It usually takes several minutes to restart the application instance; you are not able to access the application until it is finished.

The SAP xMII installation is complete. Click Finish to close the InstallShield Wizard.

See the SAP xMII Installation Guide for additional installation options.

Installing SAP xMII: Upgrading from an AS Version

Note: If you are upgrading from an ISAPI version, see Installing SAP xMII: Upgrading from an ISAPI Version. If you are installing SAP xMII for the first time, see Installing SAP xMII.

Note: If an error occurs during installation, review the install logs for more information.

Make sure you are logged on to the server machine using an Administrator account.

If you are running Windows NT, stop the Web server and IIS via the Control Panel Services applet.

OR

If you are running Windows 2000 or Windows 2003, stop the Web server and IIS via Start -> Programs -> Administrative Tools -> Services by stopping the IIS Admin Service.

This process automatically stops dependent services such as the World Wide Web Publishing service, FTP service, and SMTP service.

If you have CD media, insert the SAP xMII CD into your CD-ROM drive.

Auto-run opens the SAP xMII Installation page in your default browser.

OR

If you downloaded the software from the SAP Service Marketplace, extract the downloaded archive to a local directory and then open the xMIIInstall.htm page in the chosen directory.

Note: If you are using a CD and auto-run fails, you can open the xMIIInstall.htm page from the CD root. A link to the Sun Java 2 SDK installation is also available on this page. Once the JSDK is installed, you can run SETUP.EXE directly from the installation folder.

If you already installed the Sun JSDK 1.4.2_07 on the server, continue to step 5.

If a prior version of the Sun JDK or JRE is installed, uninstall the JDK/JRE and all references to Sun Java products including the following:

Java 2 Runtime Environment (JRE)

Java 2 Software Development Kit (SDK)

Residual folders and files under the root directory (i.e., C:\jdk1.3.1_01) or under \Program Files\Java

Registry entries (via Start | Run, enter Regedit in the Open field, select HKEY_LOCAL_MACHINE\SOFTWARE, and delete the JavaSoft folder and all its sub-directories)

Then install the Sun JSDK 1.4.2_07.

If the Sun JSDK 1.4.2_07 is not installed, download the JSDK from http://java.sun.com/products/archive/j2se/1.4.2_07/index.html. Click Download J2SDK, save the file to disk, and run the setup from the local disk.

Select the SAP xMII 11.5 installation link.
The InstallShield Wizard for SAP xMII screen appears.

If you did not stop the IIS services, a dialog appears. Click Yes to stop the service.
Another dialog appears.

Click OK to acknowledge that the IIS services were stopped successfully.

The Welcome screen for the InstallShield Wizard appears. Click Next.

Note: You may click Cancel at any time to stop the installation.

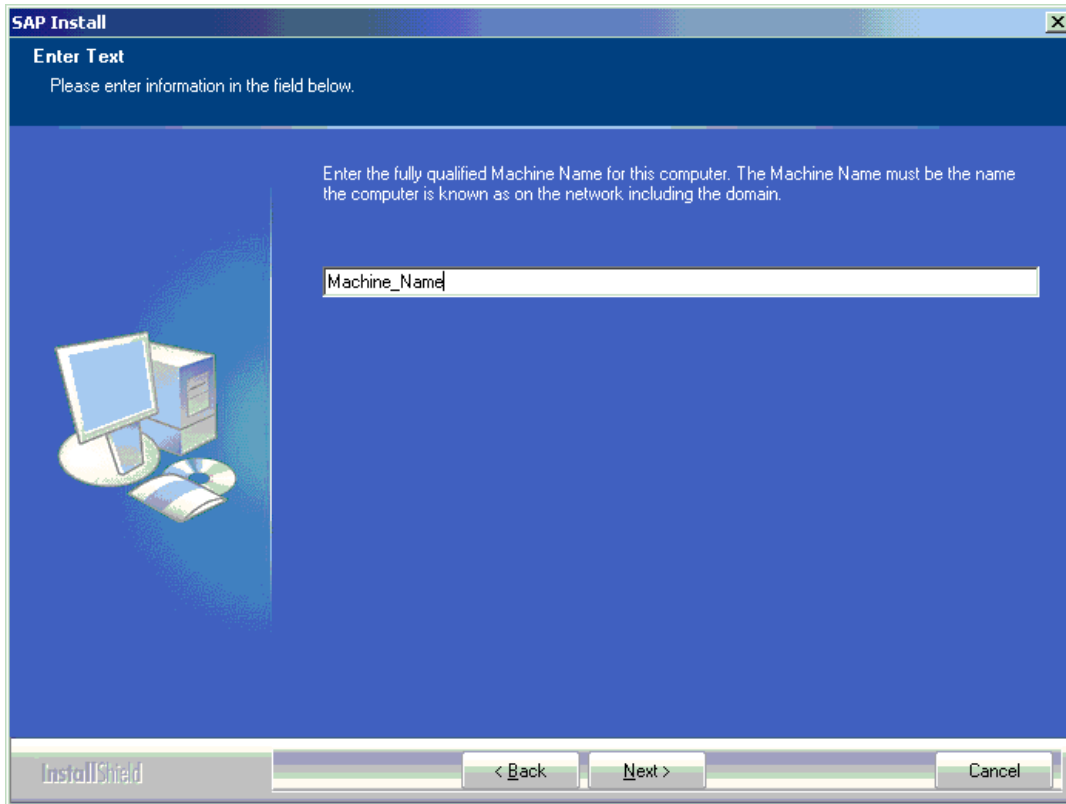
The License Agreement window appears. Read the license agreement carefully.
Click Yes to accept the agreement and continue the installation.

OR

If you do not accept the license agreement, click No. The setup stops and InstallShield Wizard closes.

Enter the fully qualified machine name (for example, MACHINE_NAME.Domain.com) of the machine on which you are installing SAP xMII. The fully qualified machine name is required to identify the location of the security server in a multiple domain and/or third-party single sign-on scenario. If you are not implementing across domains or using third-party single sign-on, the machine name (without the domain name) is sufficient.

Note: If SAP xMII is run on a port other than port 80, you must add the port number to the machine name in the SAP Install window.



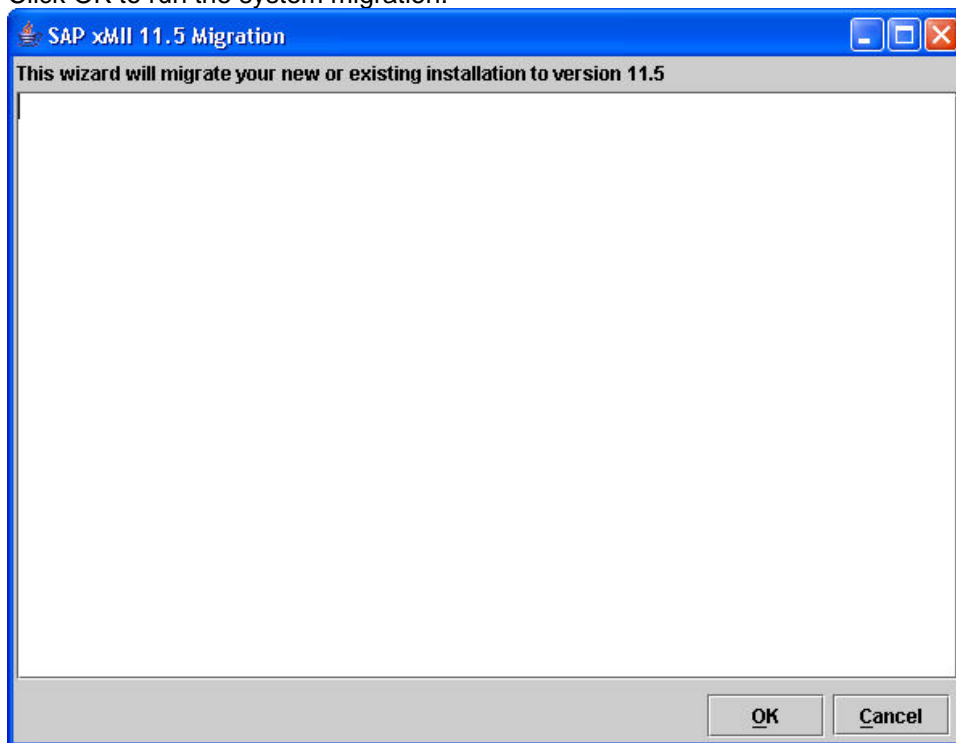
The installation begins copying files and installing. If an error occurs, the InstallShield stops and the setup does not continue. This process may take several minutes.

A Question dialog appears. Click Yes to run the version 11.5 migration tool. The migration tool upgrades your system configurations from your previous version to the current version. If you click No, you can run the migration from the Start menu after the installation.



The SAP xMII 11.5 Migration screen appears.

Click OK to run the system migration.



When the migration is complete, click Cancel.

After the installation is complete but before using SAP xMII, stop the IIS Admin Service and then restart the IIS Admin Service and the World Wide Web Publishing Service.

In Windows NT, open Windows Control Panel -> Services.

OR

In Windows 2000 and Windows 2003, choose Start -> Programs -> Administrative Tools -> Services.

Highlight IIS Admin, and click Stop to stop this service and all dependent services.

Restart the World Wide Web Publishing Service (and any other necessary Web services such as FTP or SMTP). The IIS Admin Service is automatically restarted. In Windows 2000 and Windows 2003, all Web services can be managed through Start -> Programs -> Administrative Tools -> Internet Services Manager.

Before accessing SAP xMII you also need to set up your license.

Copy your lhcms.lic license file to the C:\Lighthammer\License directory.

Restart your ServletExec-{InstanceName} (default name is ServletExec-LHCMS) NT service in the Windows Services screen.

Note: It usually takes several minutes to restart the application instance; you are not able to access the application until it is finished.

The SAP xMII installation is complete. Click Finish to close the InstallShield Wizard.

See the SAP xMII Installation Guide for additional installation options.

Installing SAP xMII: Upgrading from an ISAPI Version

Note: If you are upgrading from an AS version, see Installing SAP xMII: Upgrading from an AS Version. If you are installing SAP xMII for the first time, see Installing SAP xMII.

Note: If an error occurs during installation, review the install logs for more information.

Make sure you are logged on to the server machine using an Administrator account.

If you are running Windows NT, stop the Web server and IIS via the Control Panel Services applet.

OR

If you are running Windows 2000 or Windows 2003, stop the Web server and IIS via Start -> Programs -> Administrative Tools -> Services by stopping the IIS Admin Service.

This process automatically stops dependent services such as the World Wide Web Publishing service, FTP service, and SMTP service.

If you have CD media, insert the SAP xMII CD into your CD-ROM drive.

Auto-run opens the SAP xMII Installation page in your default browser.

OR

If you downloaded the software from the SAP Service Marketplace, extract the downloaded archive to a local directory and then open the xMIIInstall.htm page in the chosen directory.

Note: If you are using a CD and auto-run fails, you can open the xMIIInstall.htm page from the CD root. A link to the Sun Java 2 SDK installation is also available on this page. Once the JSDK is installed, you can run SETUP.EXE directly from the installation folder.

If you already installed the Sun JSDK 1.4.2_07 on the server, continue to step 5.

If a prior version of the Sun JDK or JRE is installed, uninstall the JDK/JRE and all references to Sun Java products including the following:

Java 2 Runtime Environment (JRE)

Java 2 Software Development Kit (SDK)

Residual folders and files under the root directory (i.e., C:\jdk1.3.1_01) or under \Program Files\Java

Registry entries (via Start | Run, enter Regedit in the Open field, select HKEY_LOCAL_MACHINE\SOFTWARE, and delete the JavaSoft folder and all its sub-directories)

Then install the Sun JSDK 1.4.2_07.

If the Sun JSDK 1.4.2_07 is not installed, download the JSDK from http://java.sun.com/products/archive/j2se/1.4.2_07/index.html. Click Download J2SDK, save the file to disk, and run the setup from the local disk.

Select the SAP xMII 11.5 installation link.

The InstallShield Wizard for SAP xMII screen appears.

If you did not stop the IIS services, a dialog appears. Click Yes to stop the service. Another dialog appears.

Click OK to acknowledge that the IIS services were stopped successfully.

The Welcome screen for the InstallShield Wizard appears. Click Next.

Note: You may click Cancel at any time to stop the installation.

The License Agreement window appears. Read the license agreement carefully. Click Yes to accept the agreement and continue the installation.

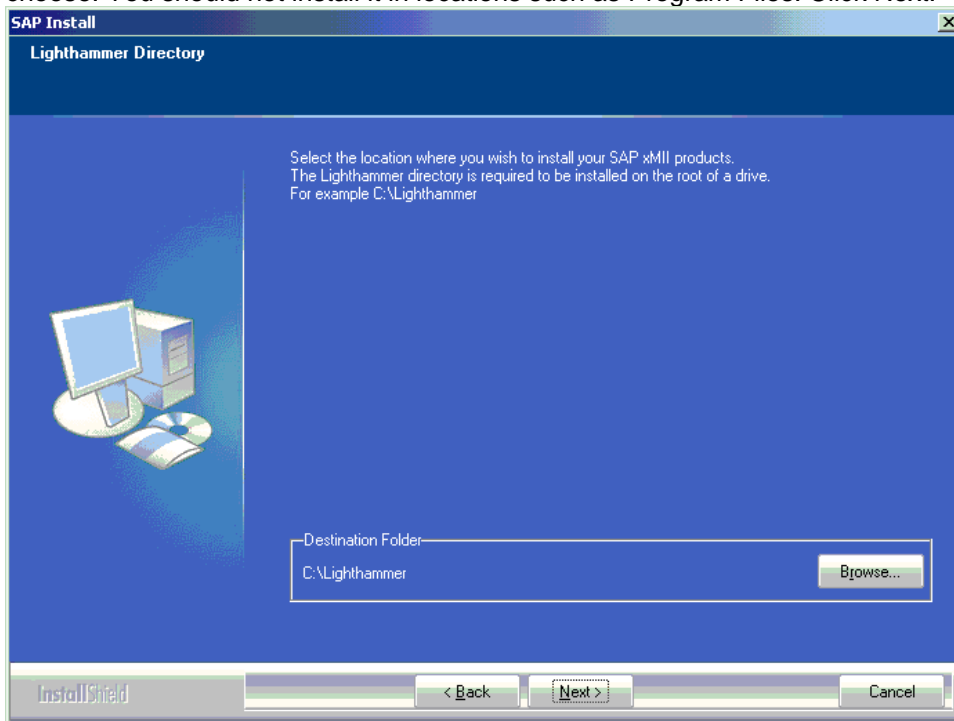
OR

If you do not accept the license agreement, click No. The setup stops and InstallShield Wizard closes.

If you are upgrading from 11.0.x ISAPI, continue to step 12.

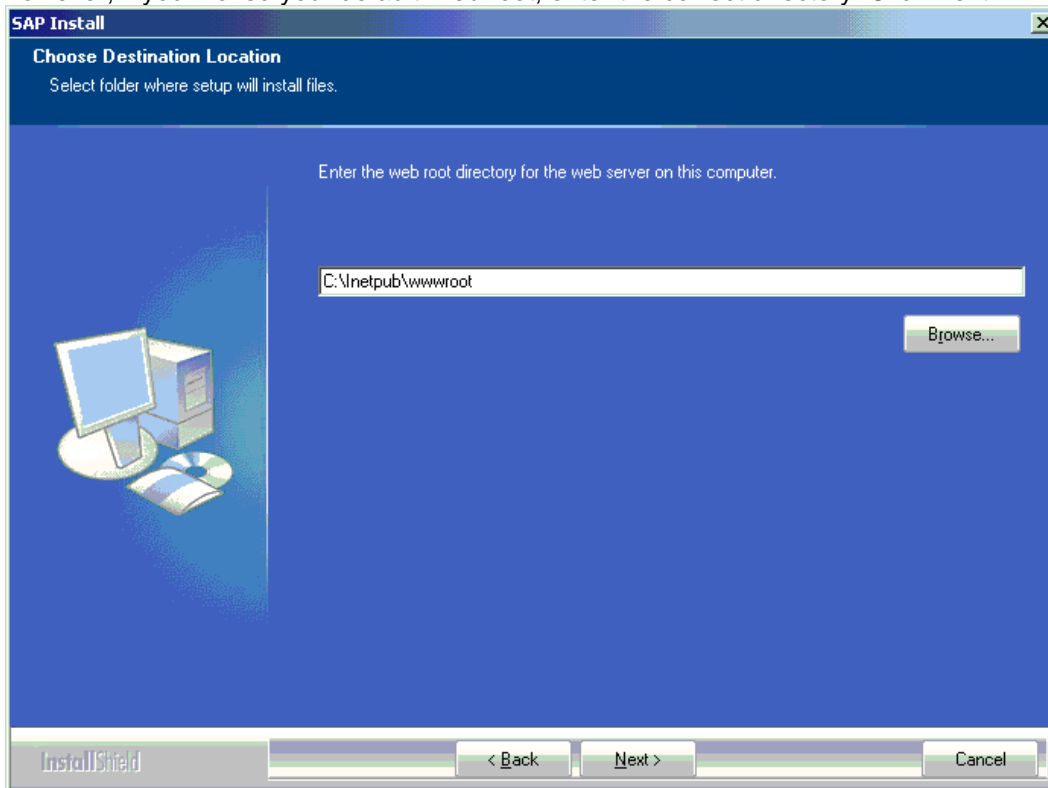
If you are upgrading from 10.1.x or prior, the Lighthammer Directory screen appears.

Choose the destination folder where you want to install the SAP xMII application. The default destination folder is acceptable, although it is recommended that you install the Lighthammer directory to the same drive on which you installed IIS. The Lighthammer directory should be installed on the root of the drive you choose. You should not install it in locations such as Program Files. Click Next.



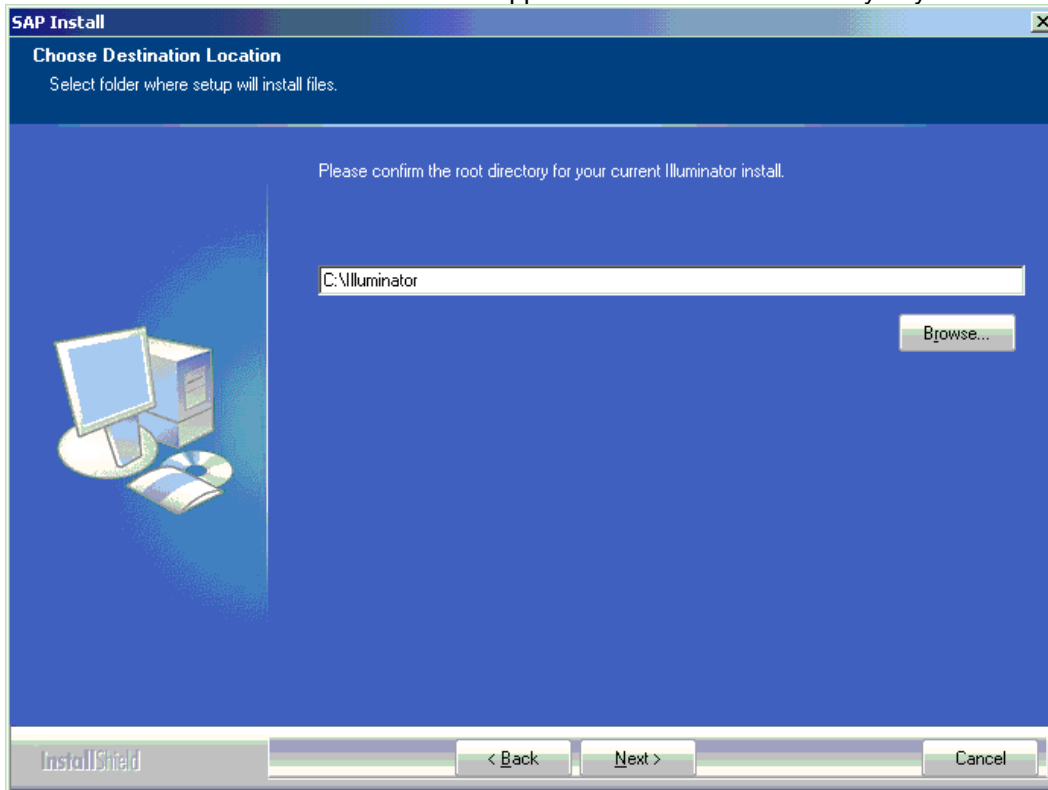
The Confirm New Folder dialog appears. Click Yes to create the new folder.

The Choose Destination Location screen appears. The default location of C:\inetpub\wwwroot is acceptable; however, if you moved your default Web root, enter the correct directory. Click Next.



A Question dialog appears to verify that a version prior to 11.0 was found on your system. Click Yes to continue the installation.

The Choose Destination Location screen appears. Enter the root directory of your current installation.



The ServletExec ISAPI filter must be uninstalled in order to continue with this installation. To correctly uninstall the ISAPI filter, the installation must first reinstall the ISAPI filter. Click OK after reading the informational dialog.

The ServletExec ISAPI Setup screen appears.

Click Next in all ISAPI setup screens to accept all defaults.

Note: Ignore any errors that may occur.

Clear the Display the README file check box. You do not need to read the file.

Click Finish.

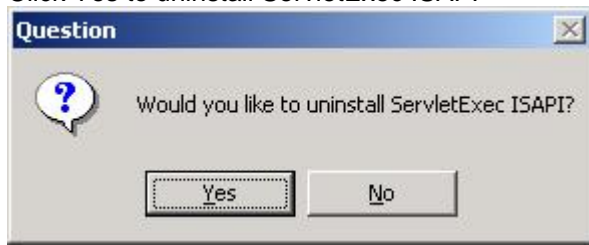
Close the ServletExec 5.0 ISAPI screen that appears.

Once the ISAPI install completes, you must uninstall it.

A dialog appears. Click OK to uninstall the ISAPI filter.

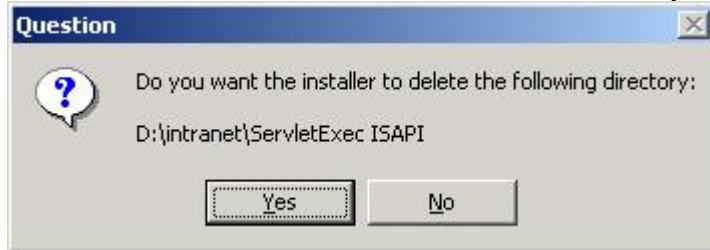


Click Yes to uninstall ServletExec ISAPI



Click Yes to stop the IIS Admin service.

Click Yes to delete the ServletExec ISAPI install directory.



A dialog appears that states the uninstall is complete. Click OK.

After the ISAPI uninstall is complete, the installation continues.

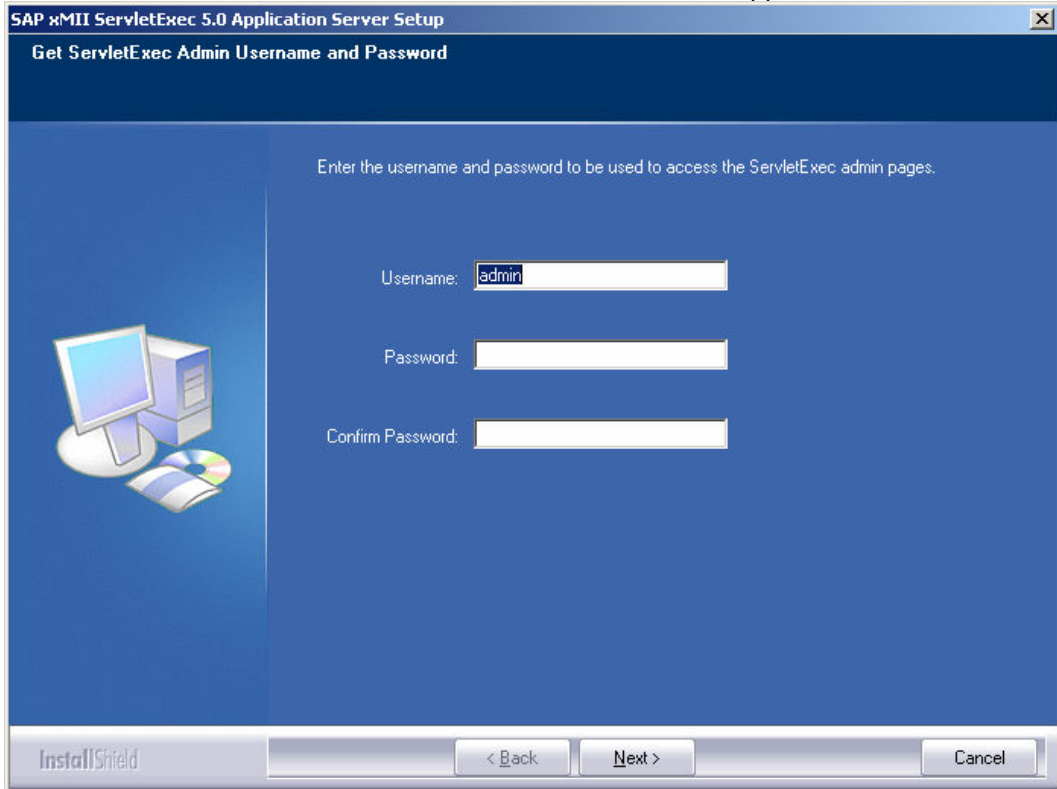
The Choose Destination Location screen appears. Click Next to set the destination folder as the default and initiate the ServletExec AS installation.

It may take several minutes. You may see one or more windows open and close during this process.

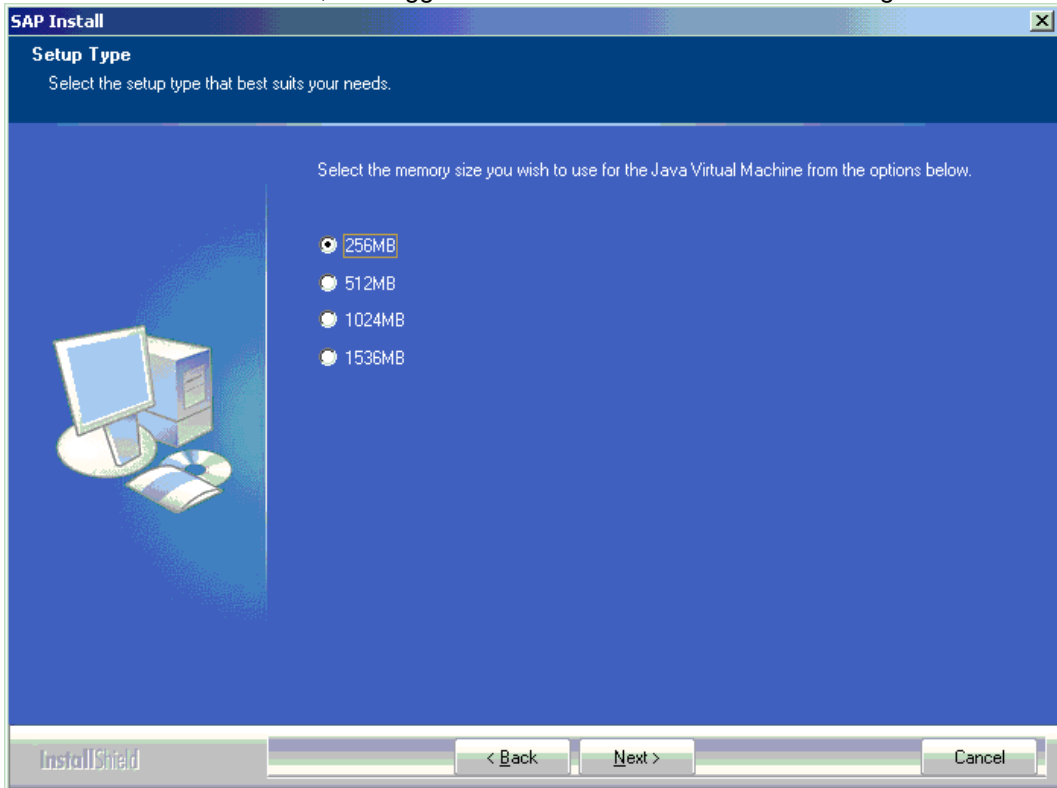
Wait for the ServletExec AS install to complete.

Click Yes to stop the IIS Admin Service.

The Get ServletExec Admin Username and Password screen appears. Enter the username and password.



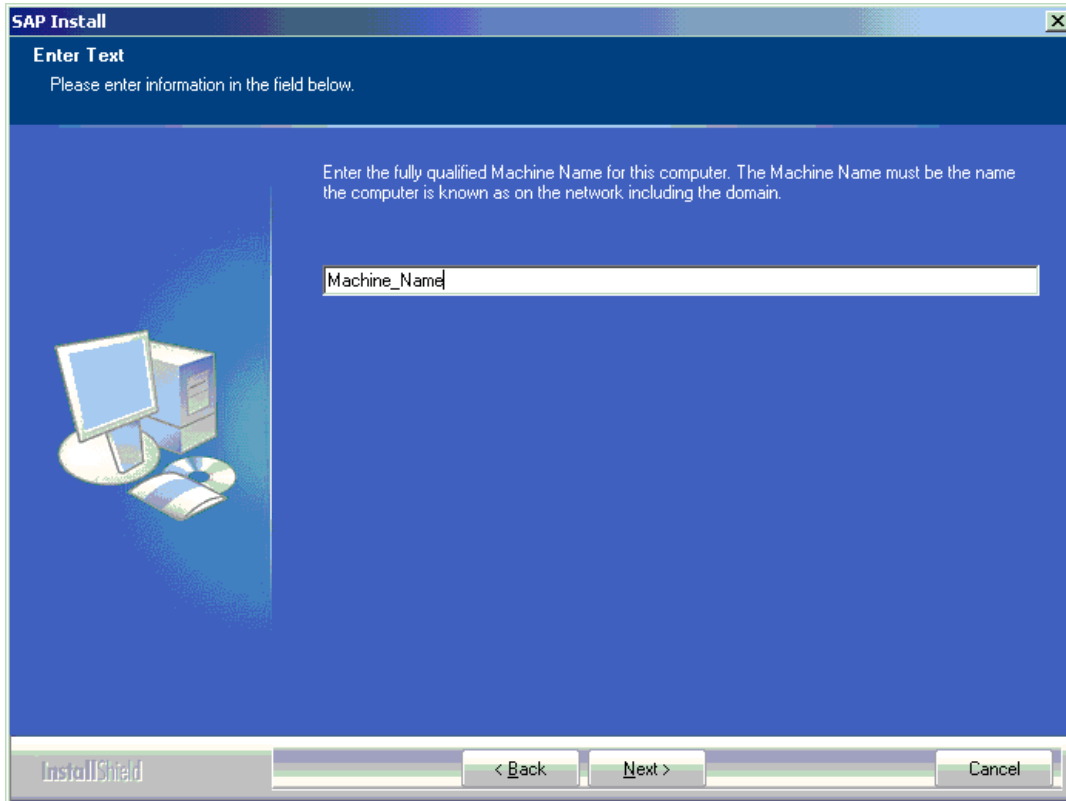
The Setup Type screen appears. Select the appropriate option button for the amount of RAM you want to designate to SAP xMII via ServletExec AS. It is suggested that a minimum of 512 MB be selected. If it is a dedicated SAP xMII server, we suggest one-half of the server's RAM as a guideline.



Note: Once the installation is complete and SAP xMII is operating properly on ServletExec Application Server, it is no longer necessary to restart IIS in order to restart SAP xMII. Go to the Services console, and restart the ServletExec-<InstanceName> (default name is ServletExec-LHCMS) service. This process reloads SAP xMII.

Enter the fully qualified machine name (for example, MACHINE_NAME.Domain.com) of the machine on which you are installing SAP xMII. The fully qualified machine name is required to identify the location of the security server in a multiple domain and/or third-party single sign-on scenario. If you are not implementing across domains or using third-party single sign-on, the machine name (without the domain name) is sufficient.

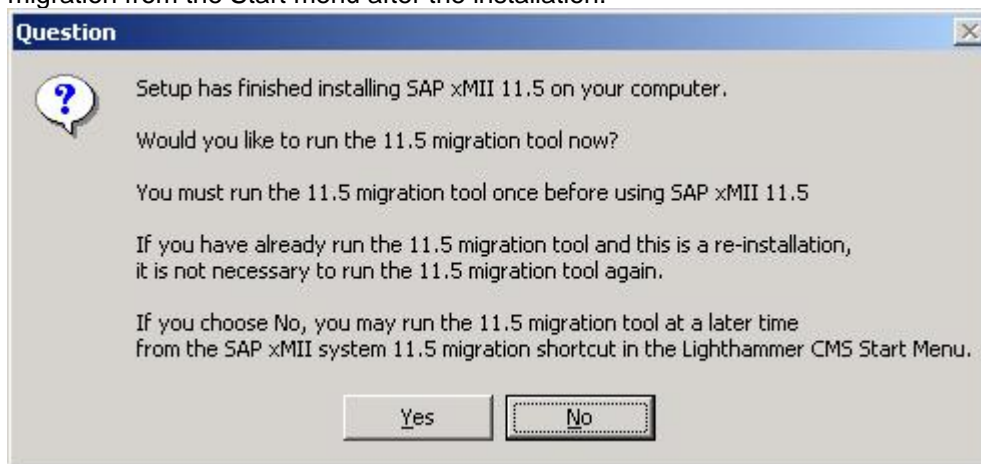
Note: If SAP xMII is run on a port other than port 80, you must add the port number to the machine name in the SAP Install window.



The installation begins copying files and installing. If an error occurs, the InstallShield stops and the setup does not continue. This process may take several minutes.

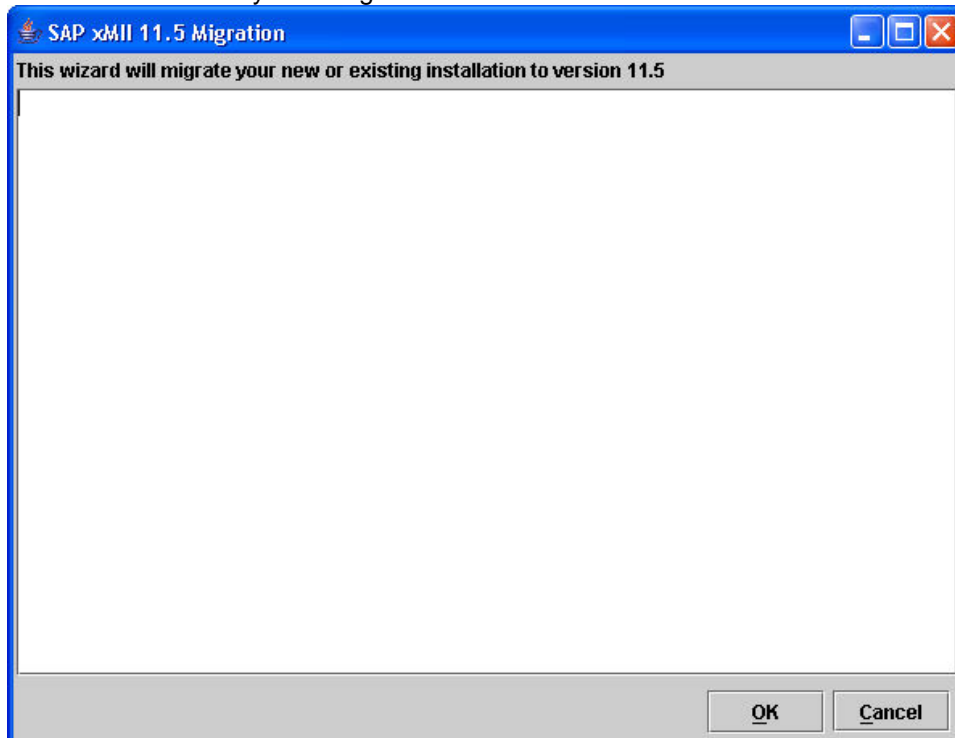
A Question dialog appears. Click Yes to run the version 11.5 migration tool. The migration tool upgrades your system configurations from your previous version to the current version. If you click No, you can run the

migration from the Start menu after the installation.



The SAP xMII 11.5 Migration screen appears.

Click OK to run the system migration.



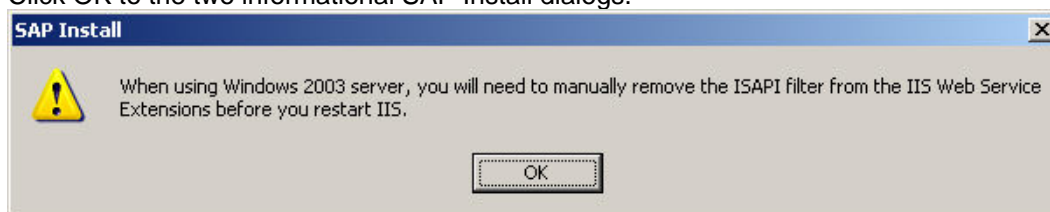
When the migration is complete, click Cancel.

If you are upgrading from 11.0.x, continue to step 22.

If you are upgrading from 10.1.x or prior, a Question dialog appears. Click Yes to run the Security migration tool.



Click OK to the two informational SAP Install dialogs.



Click Finish. The installation is complete.

See the SAP xMII Installation Guide for additional installation options.

Applet Introduction

Applet Overview

SAP xApp Manufacturing Integration and Intelligence (SAP xMII) graphical user interface components consist of a number of applets that provide a rich set of data display capabilities for a wide range of data sources. The applets are capable of a broad range of display modes, including a variety of business charts, an intelligent grid, indicator lights, a dynamic scoreboard, a real-time ticker, and intelligent tree/list box

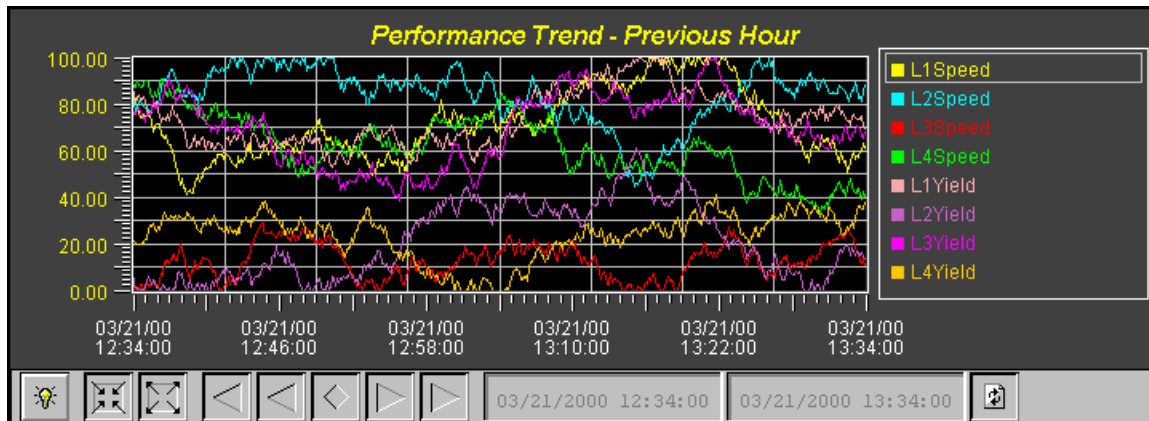
controls. An additional utility component provides a calendar applet that can be used to provide customized date selection within a Web page.

These applets provide a visual front-end to the wide-ranging server-side connector architecture that SAP xMII provides. Through a variety of techniques, the applets are "dynamically bound" to data streams from the SAP xMII connectors, providing current, historical, statistical, and other views of the broad range of plant and production data sources found in a typical manufacturing enterprise.

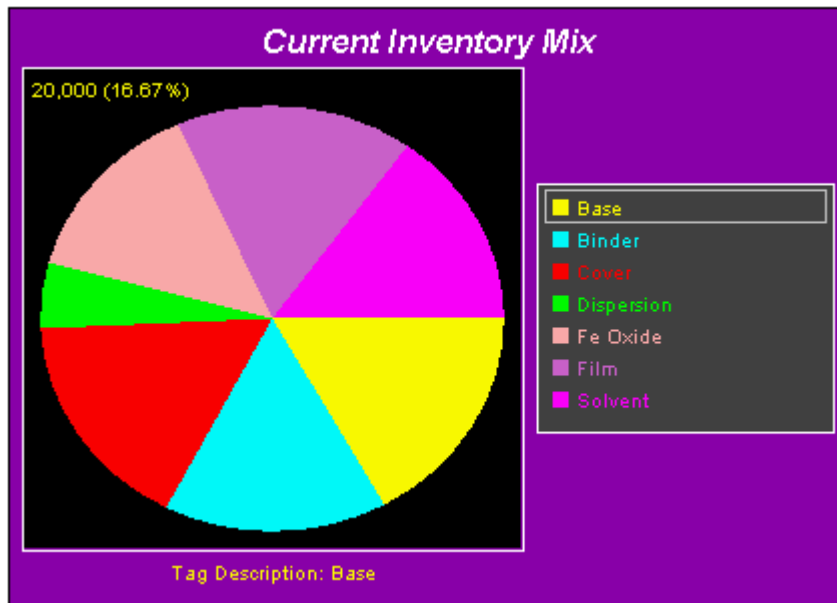
The following section provides a visual overview of the various applets provided with SAP xMII. By combining and linking these building blocks, incredibly powerful decision support applications that bridge a variety of disparate data sources can be quickly created and deployed on thin client platforms such as Web browser, personal digital assistants (PDAs), or other Web-aware mobile devices and internet appliances.

See the Applet General Reference in the xMII help documentation for more information on the applets use.

iChart Applets



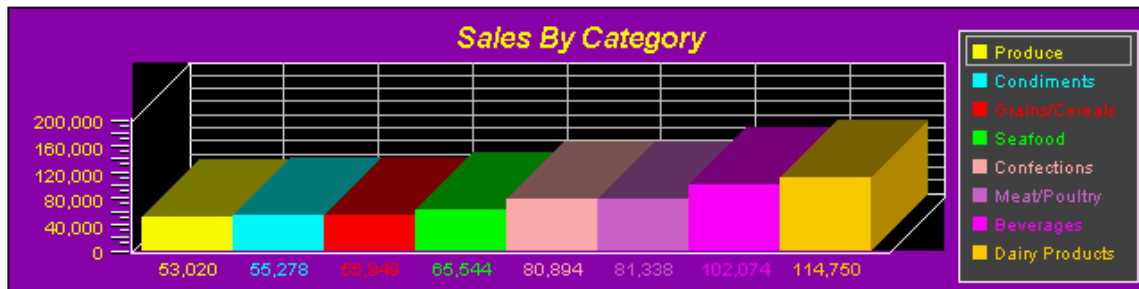
Line Mode



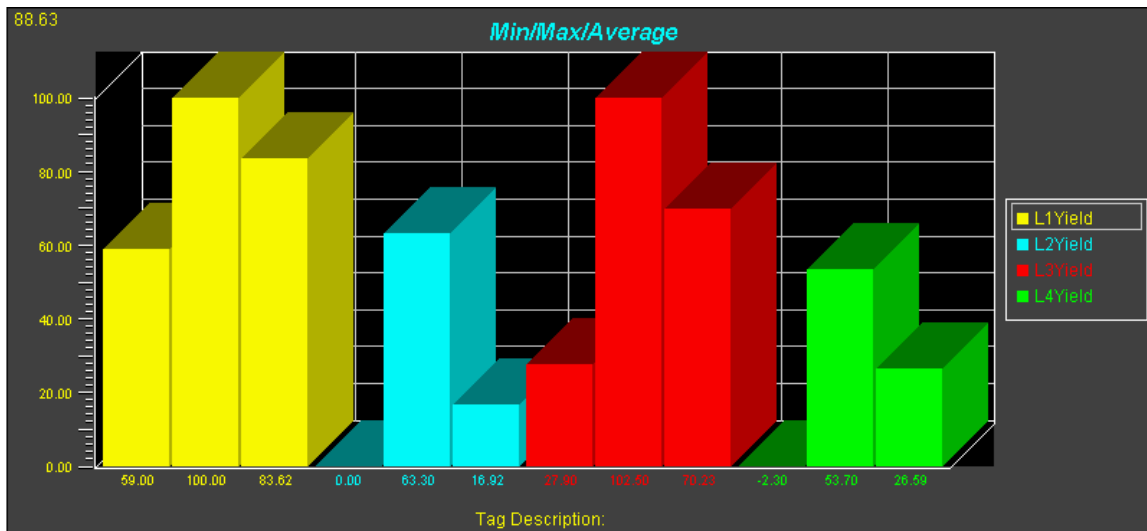
Pie Mode



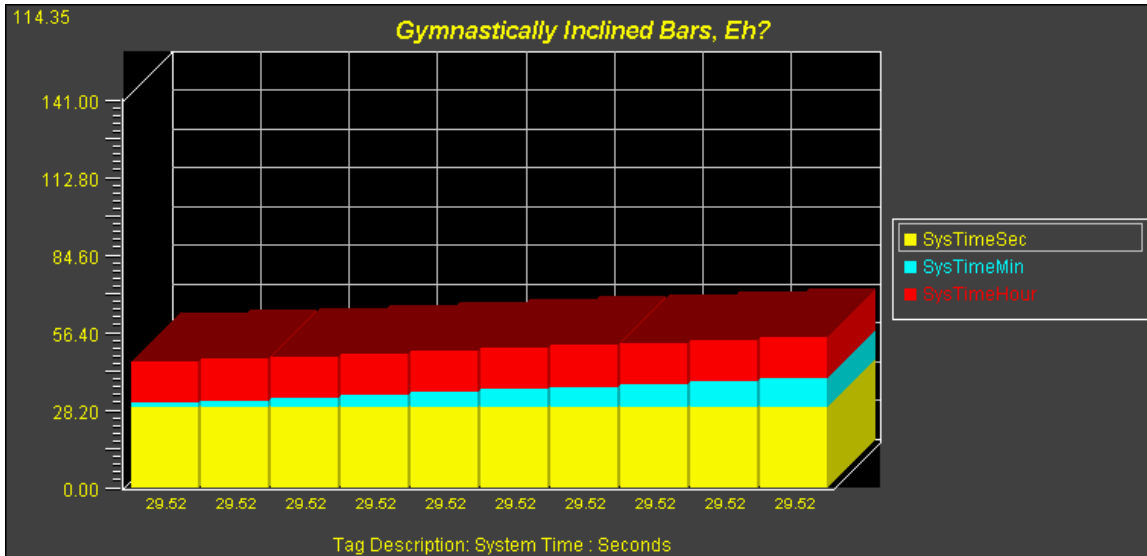
Gauge Mode



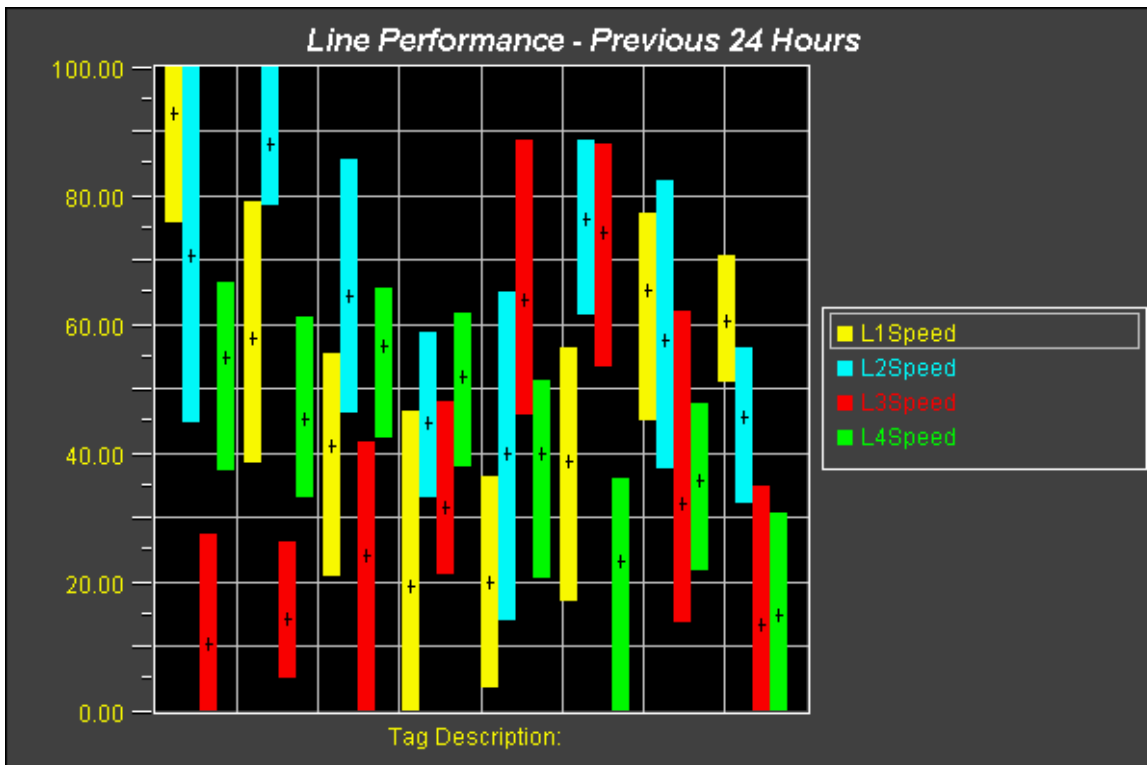
Bar Mode



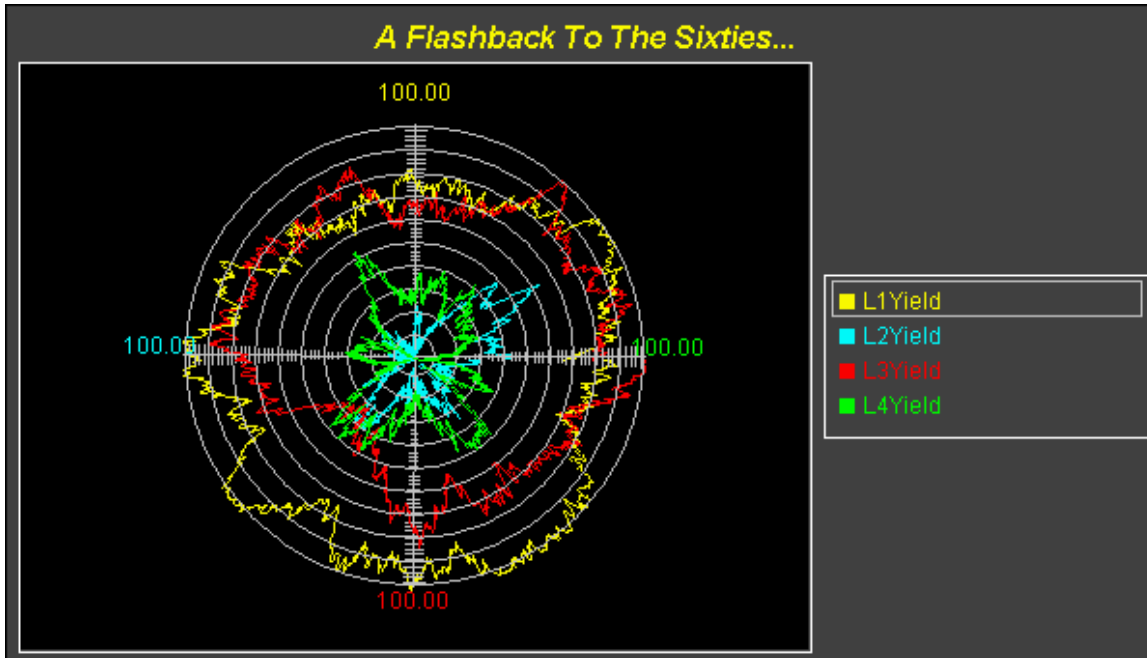
GroupBar Mode



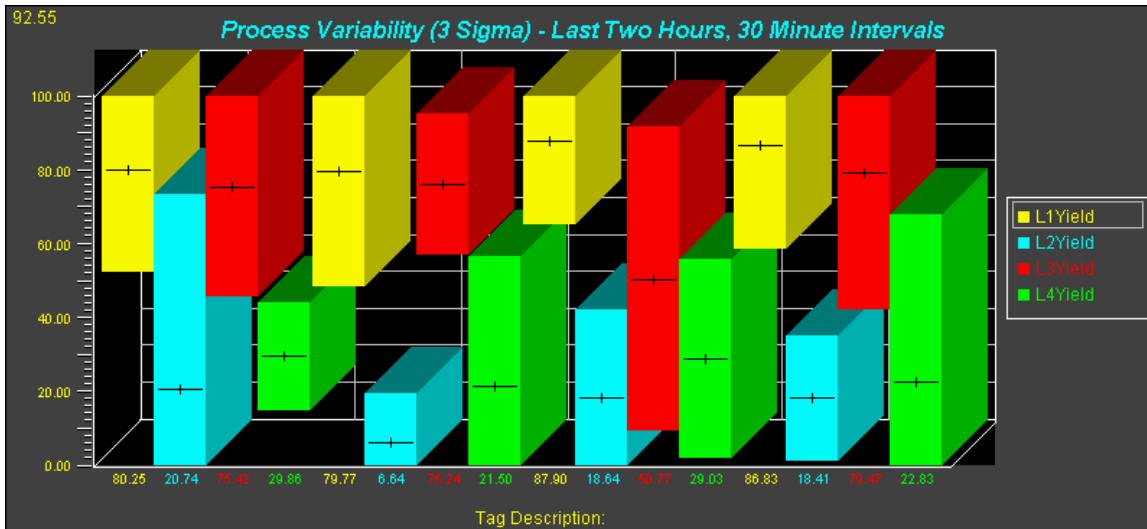
StackedBar Mode



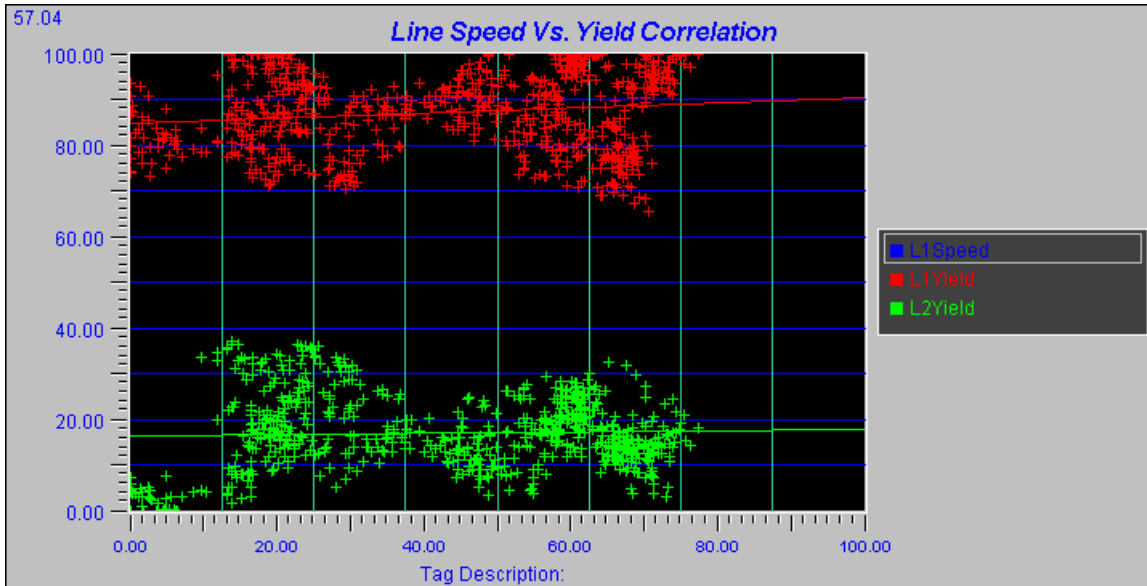
FloatingBar Mode



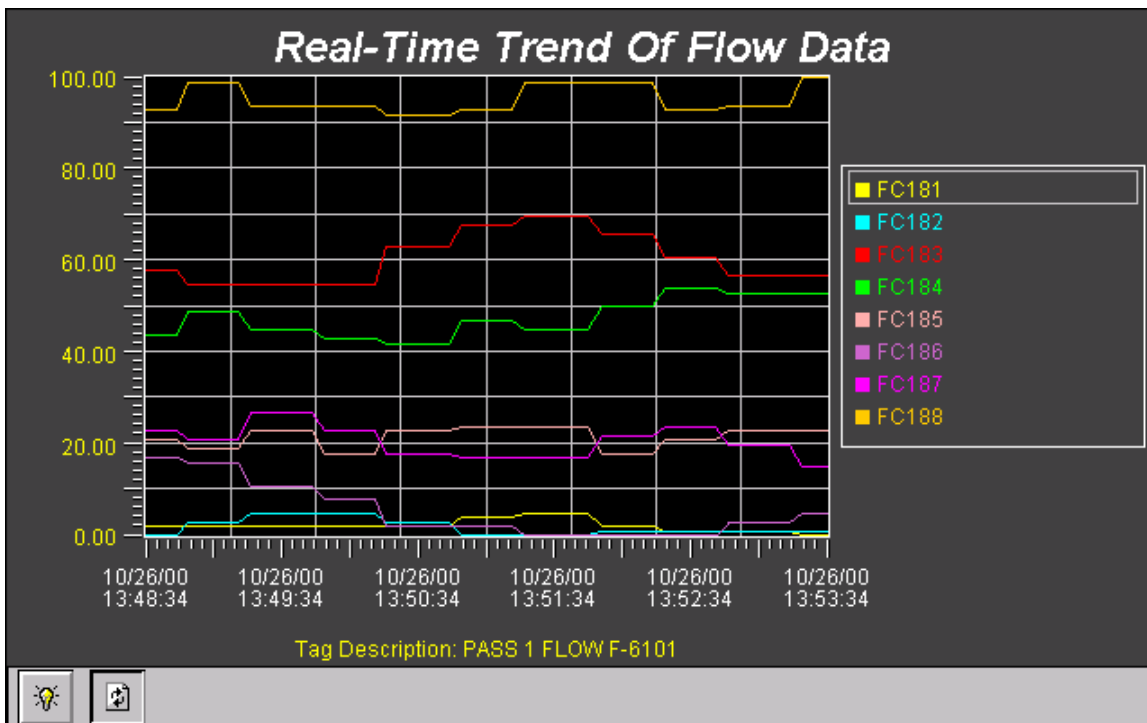
Polar Mode



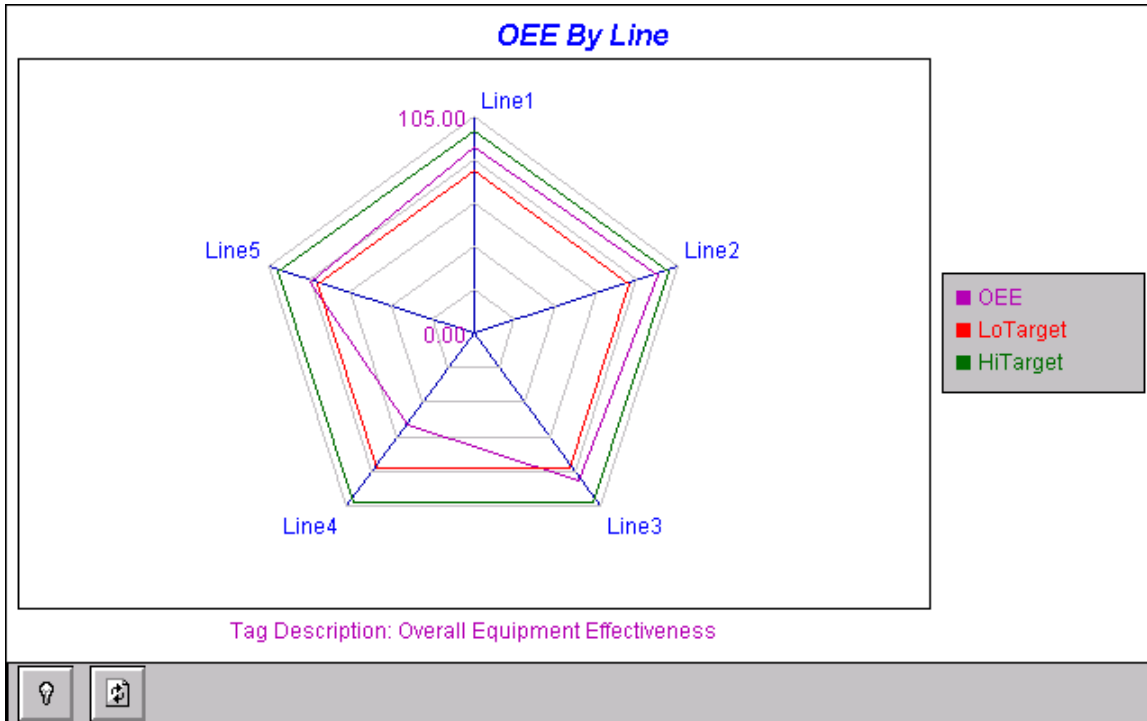
Variability Mode



XY Mode (with optional regression line)



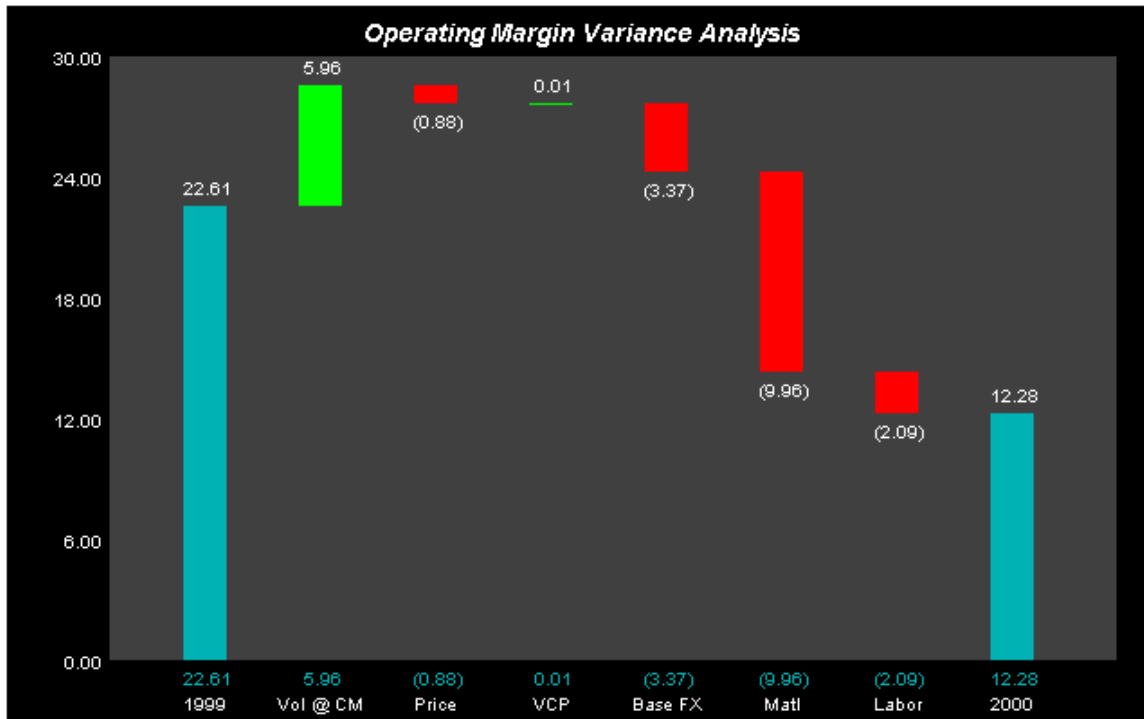
Strip Mode



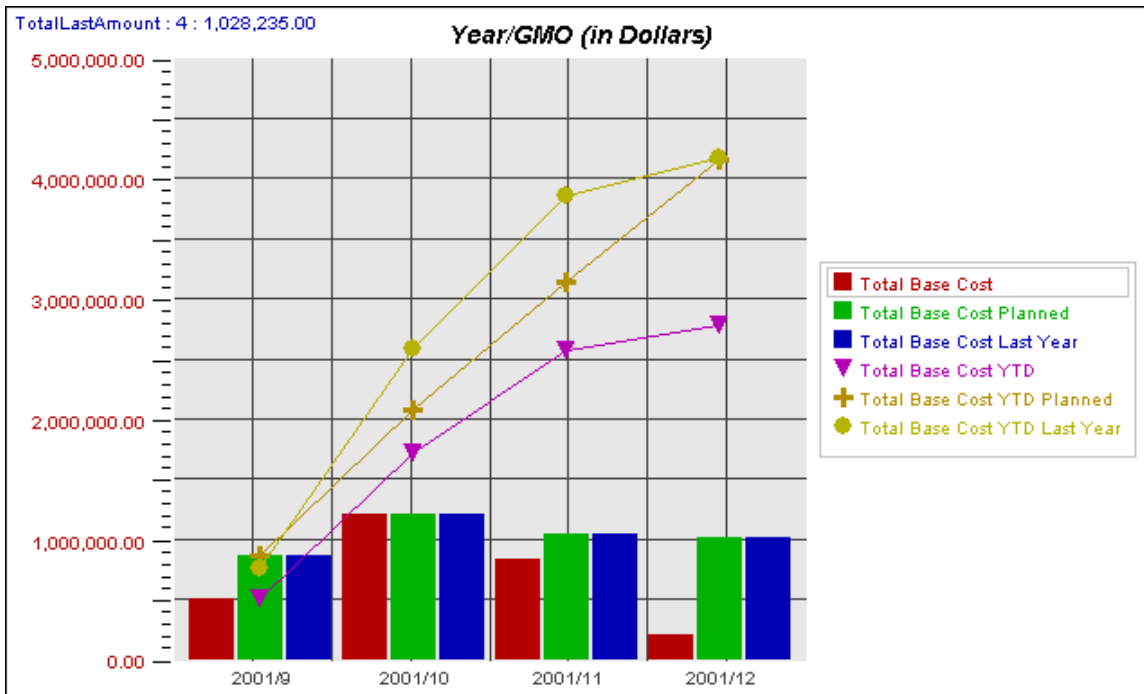
Radar Mode



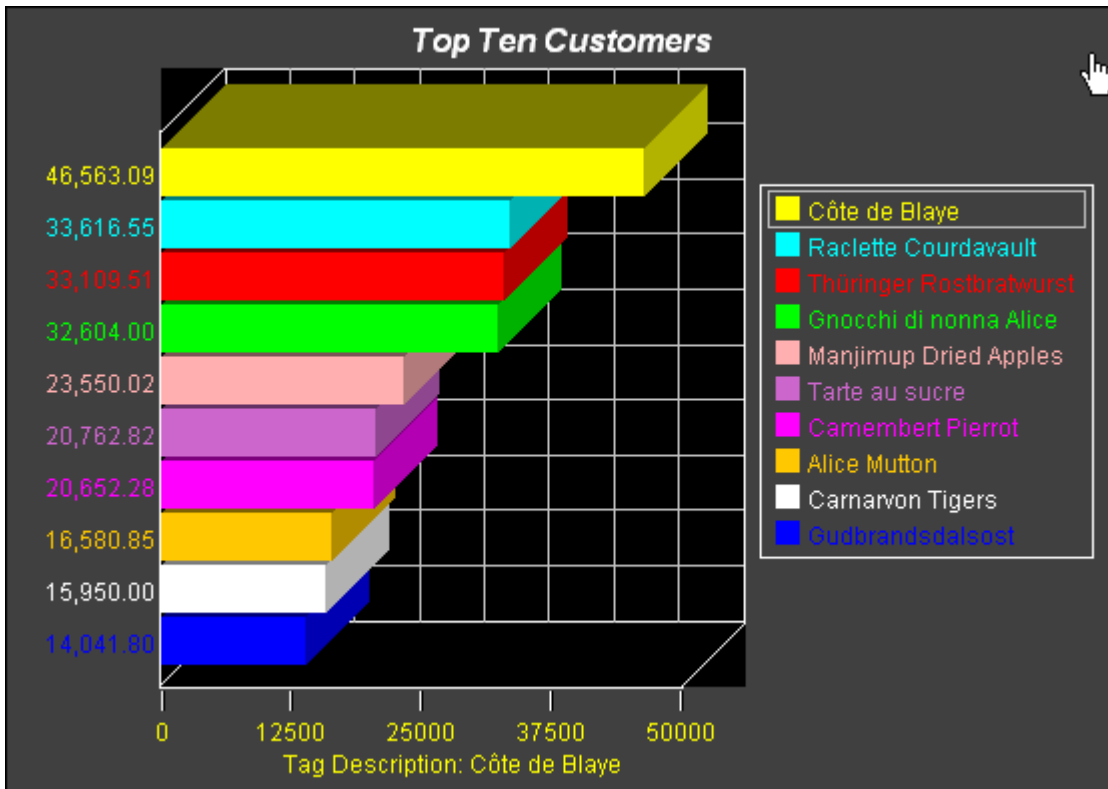
EventHorizon Mode



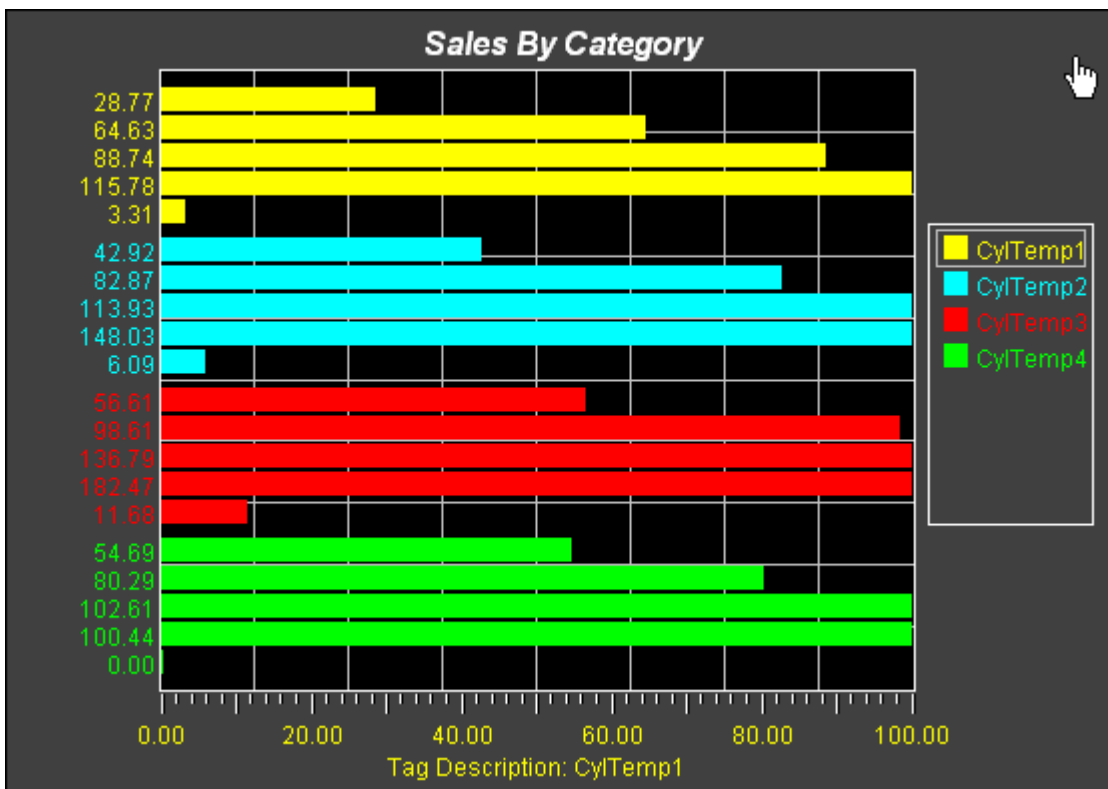
Waterfall Chart



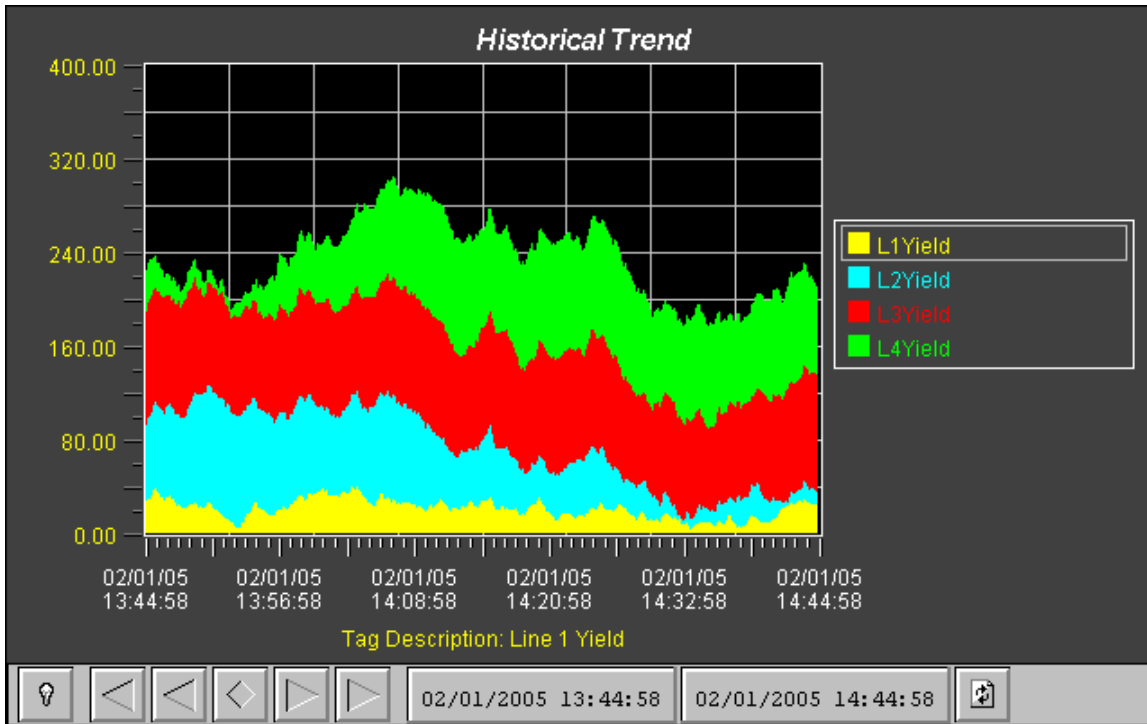
Custom Chart



Horizontal Bar Chart

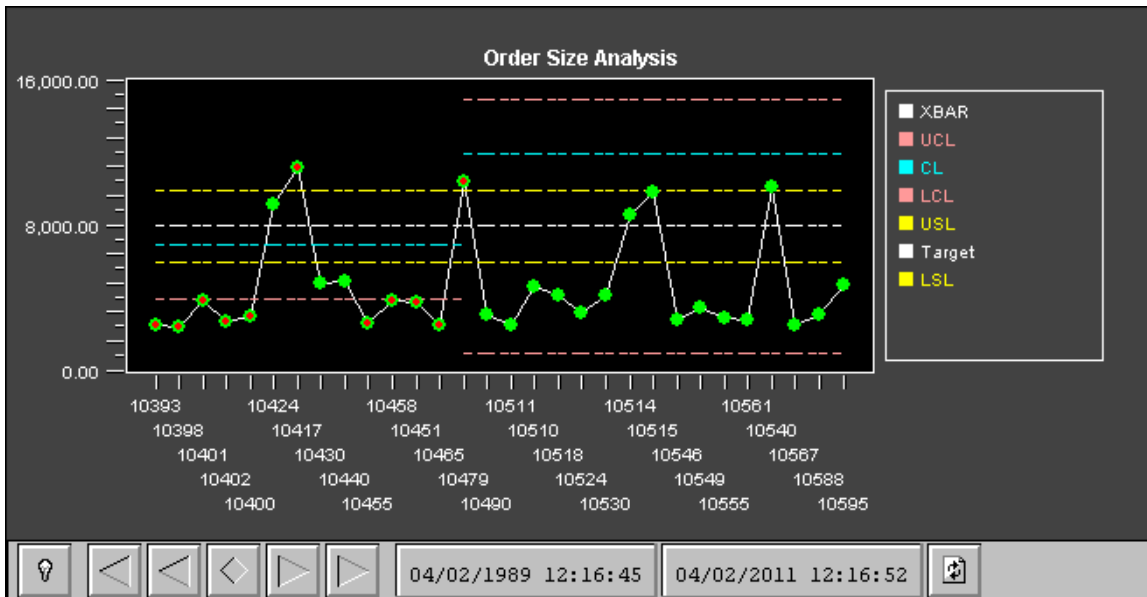


Horizontal Group Bar

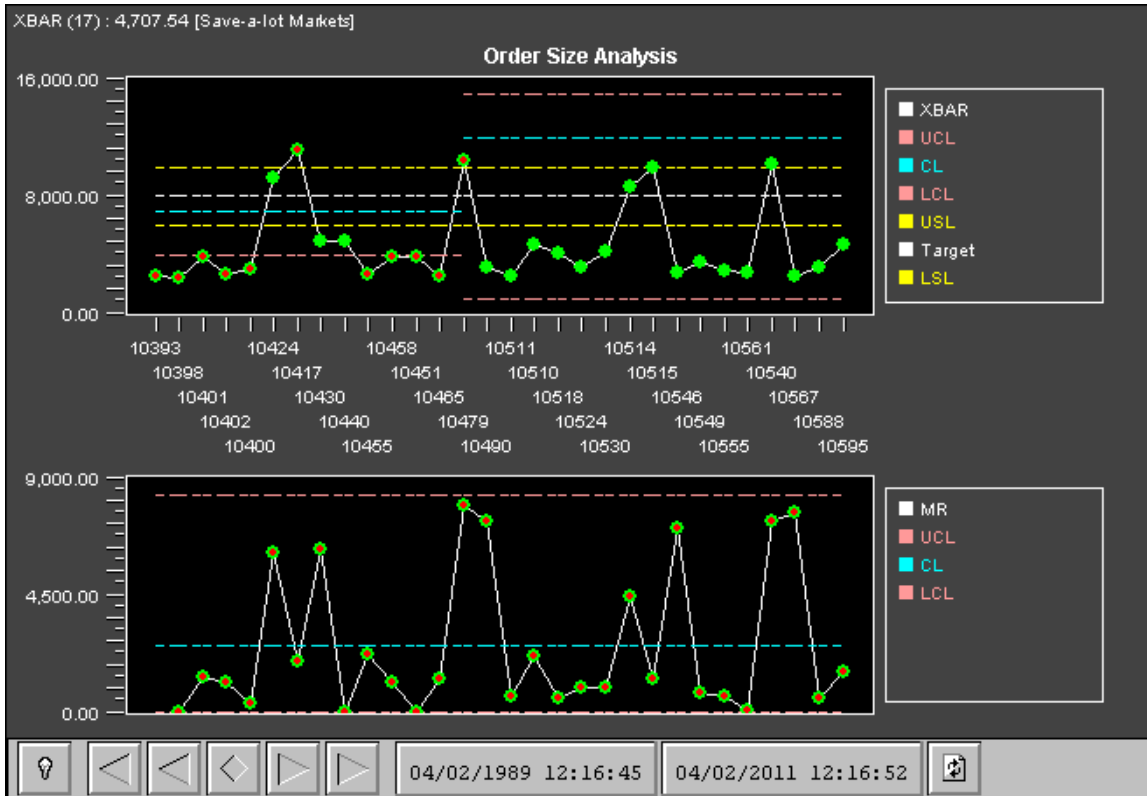


Area Chart

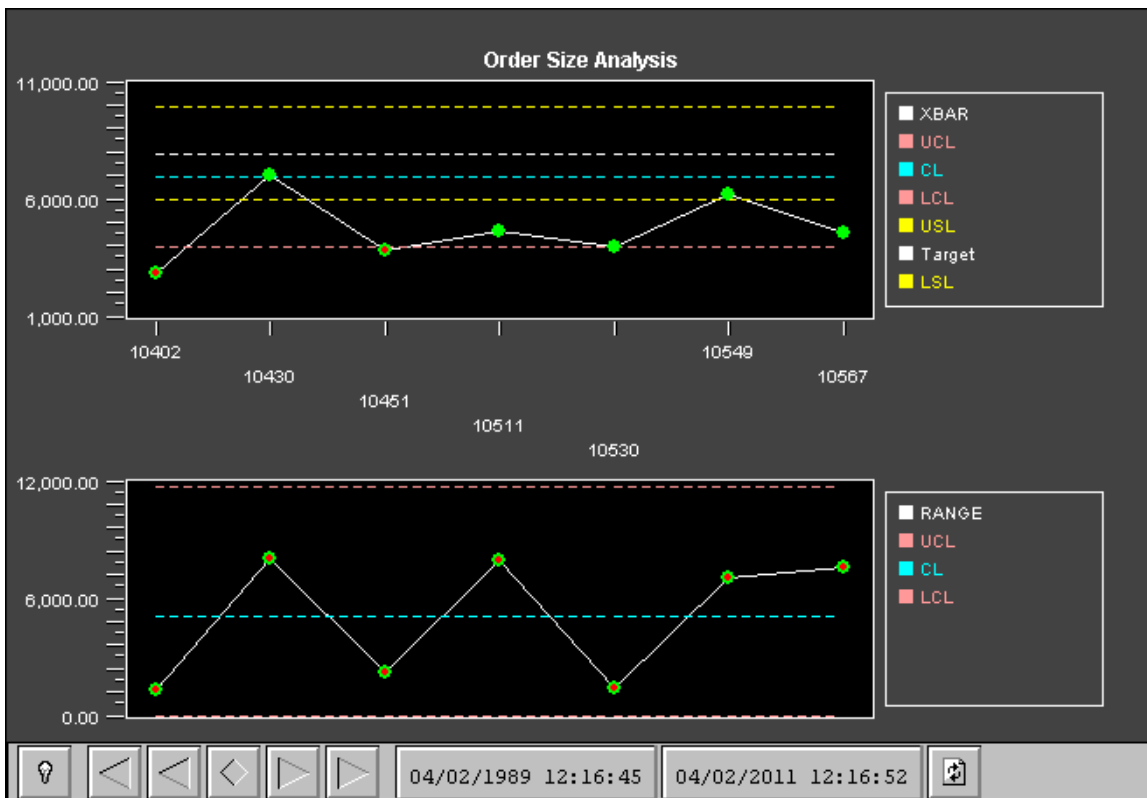
iSPCChart Applet



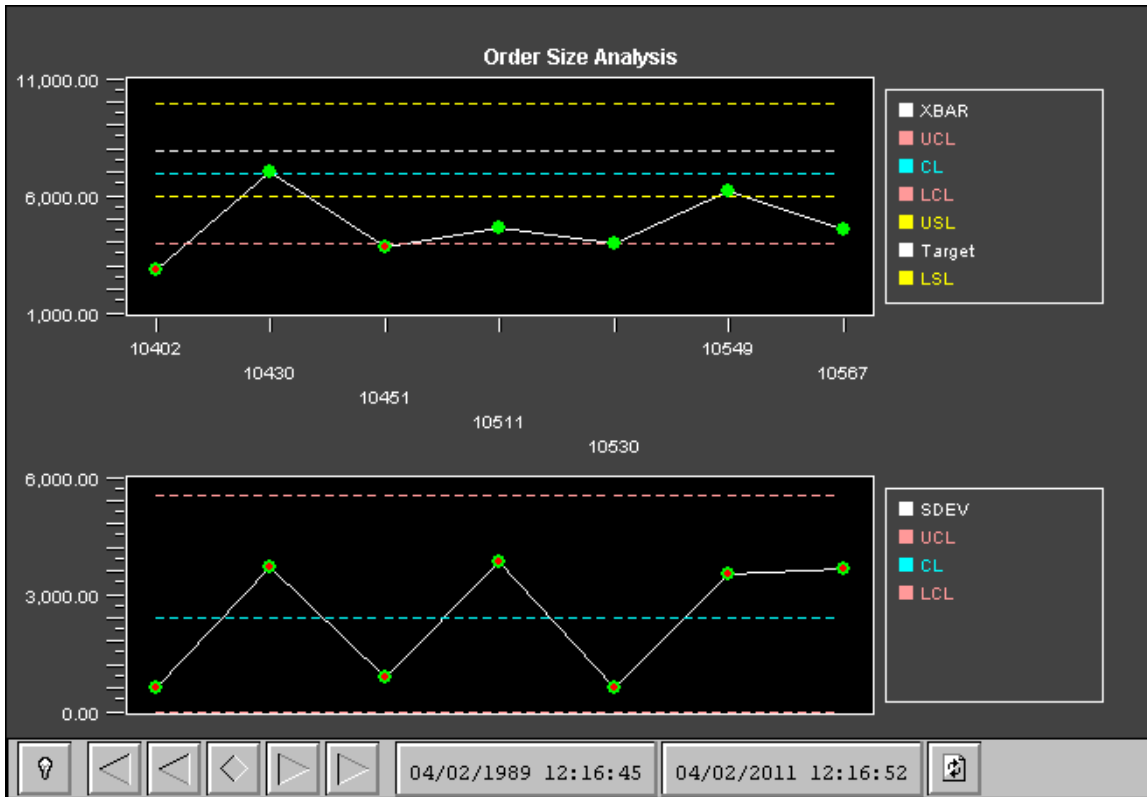
XBAR



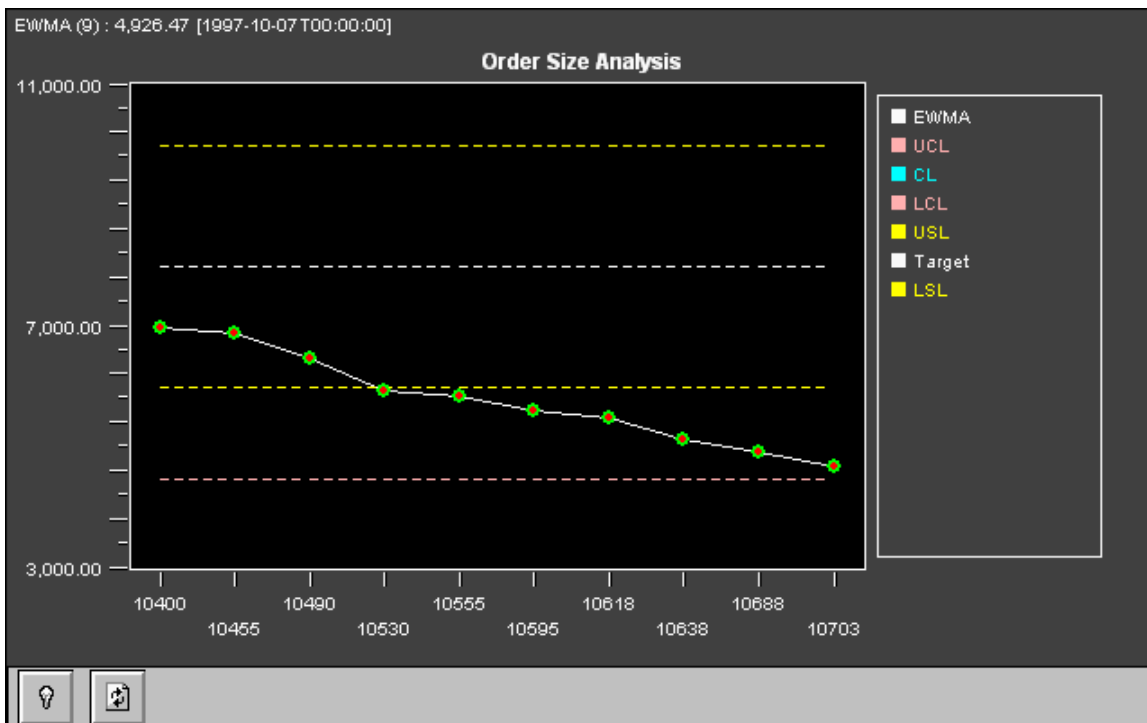
XBAR-MR



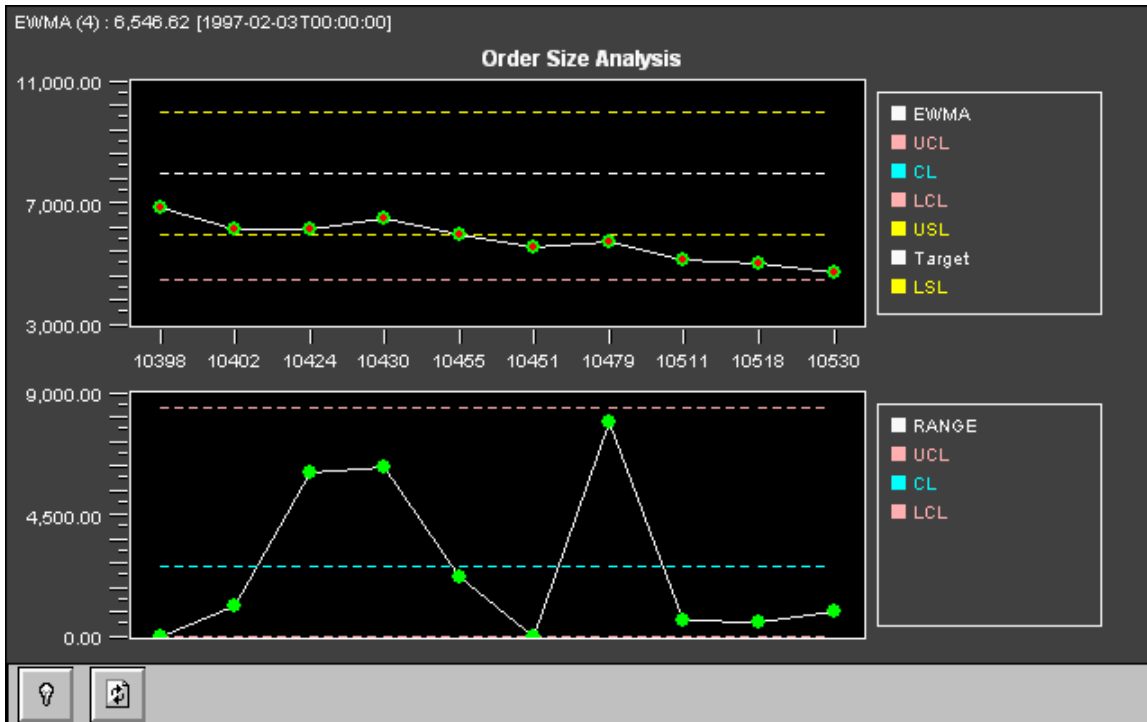
XBAR-RANGE



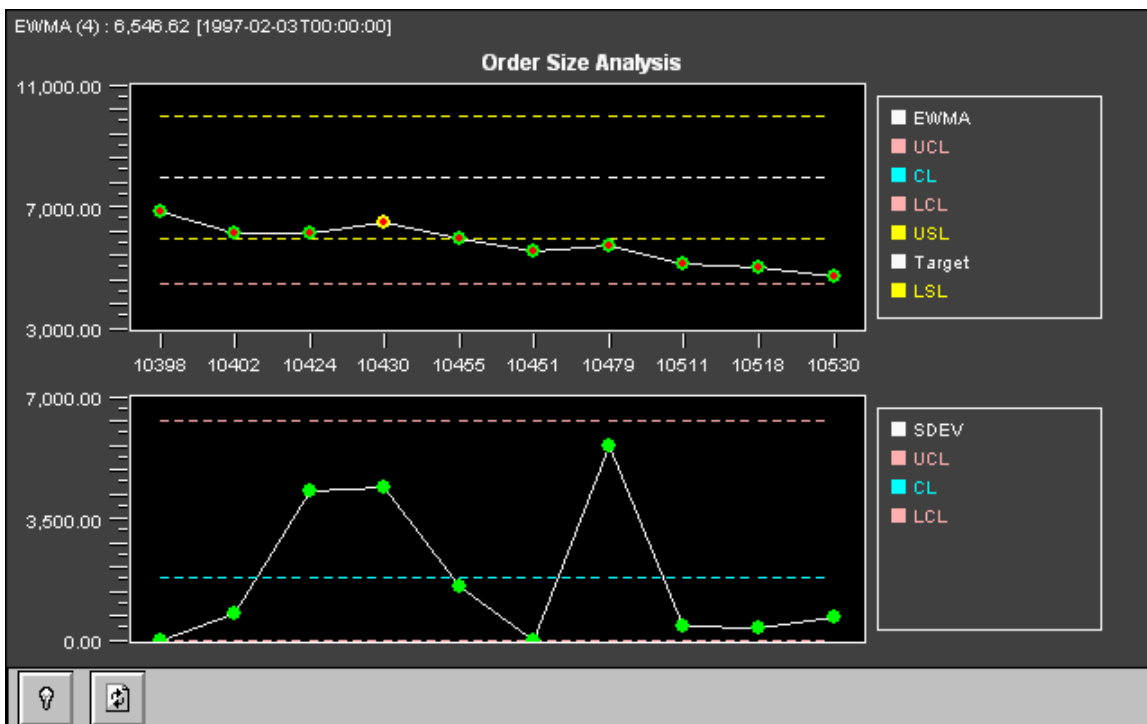
XBAR-SDEV



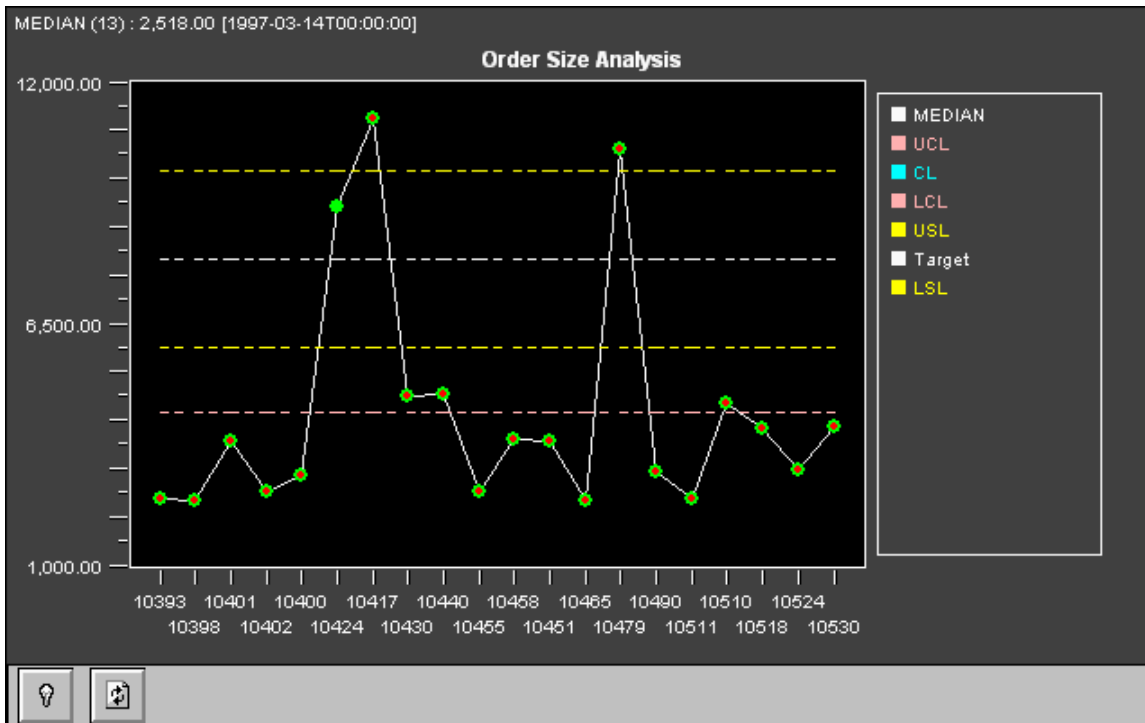
EWMA



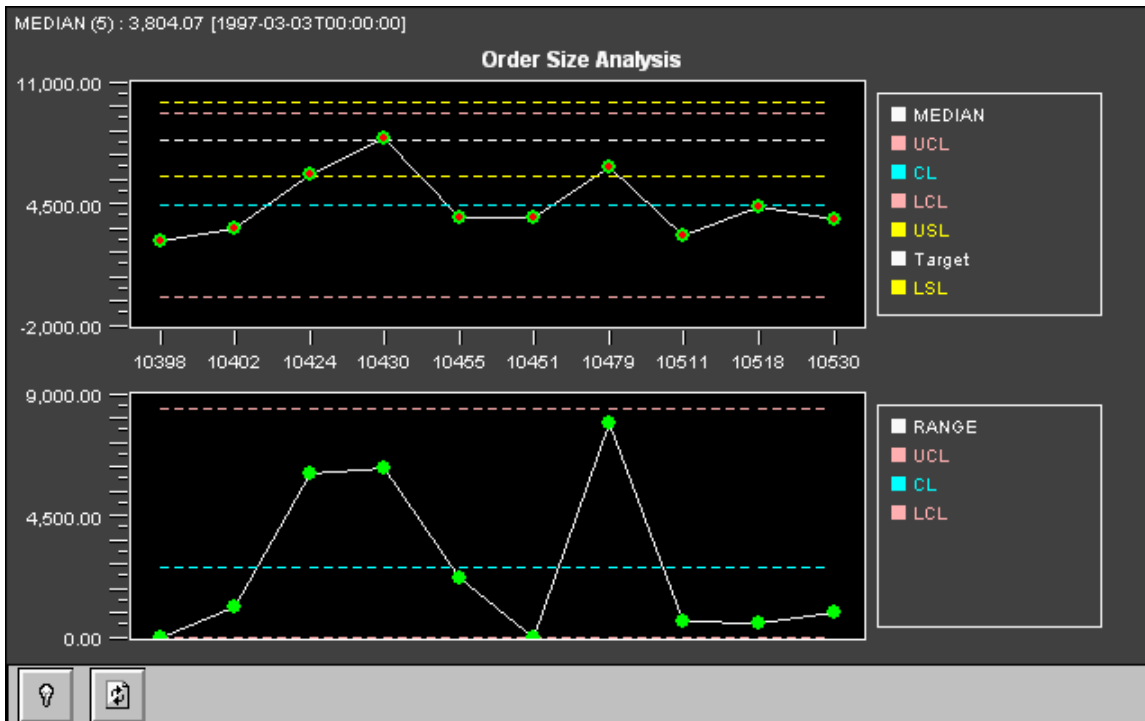
EWMA-RANGE



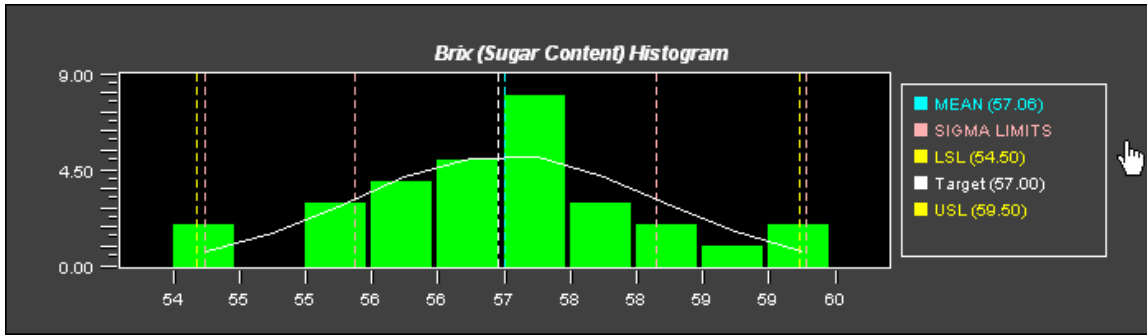
EWMA-SDEV



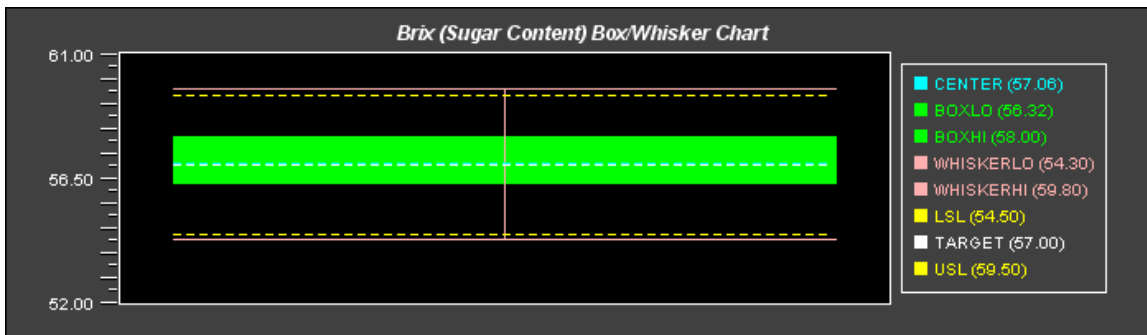
MEDIAN



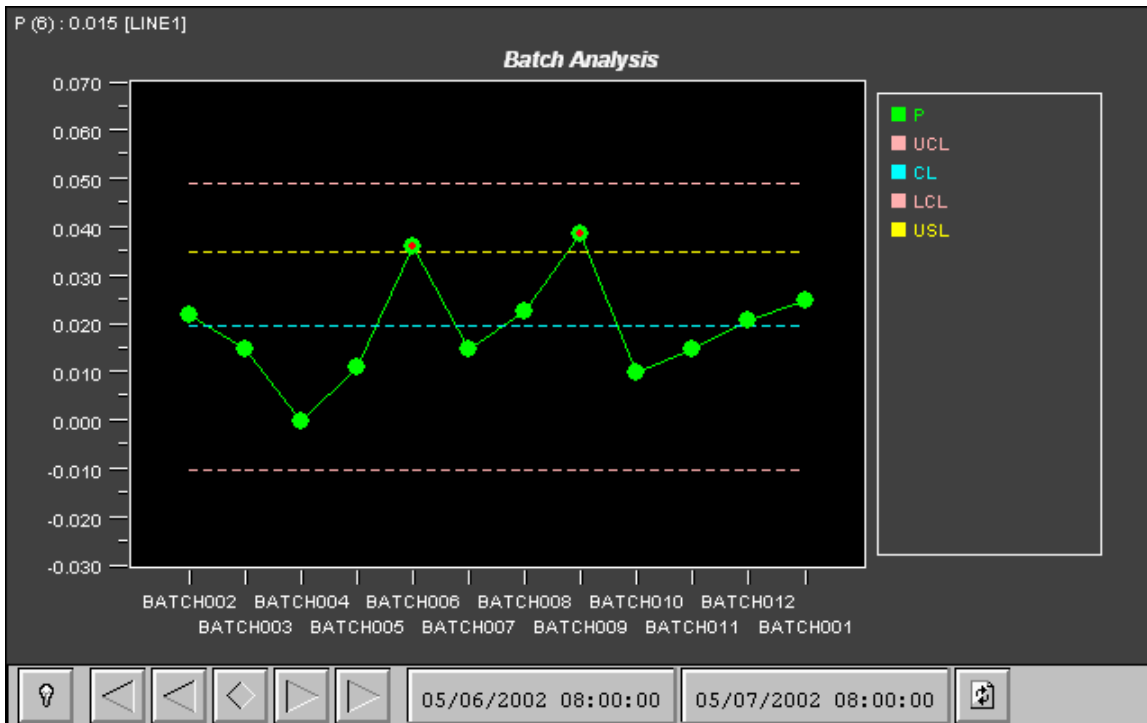
MEDIAN-RANGE



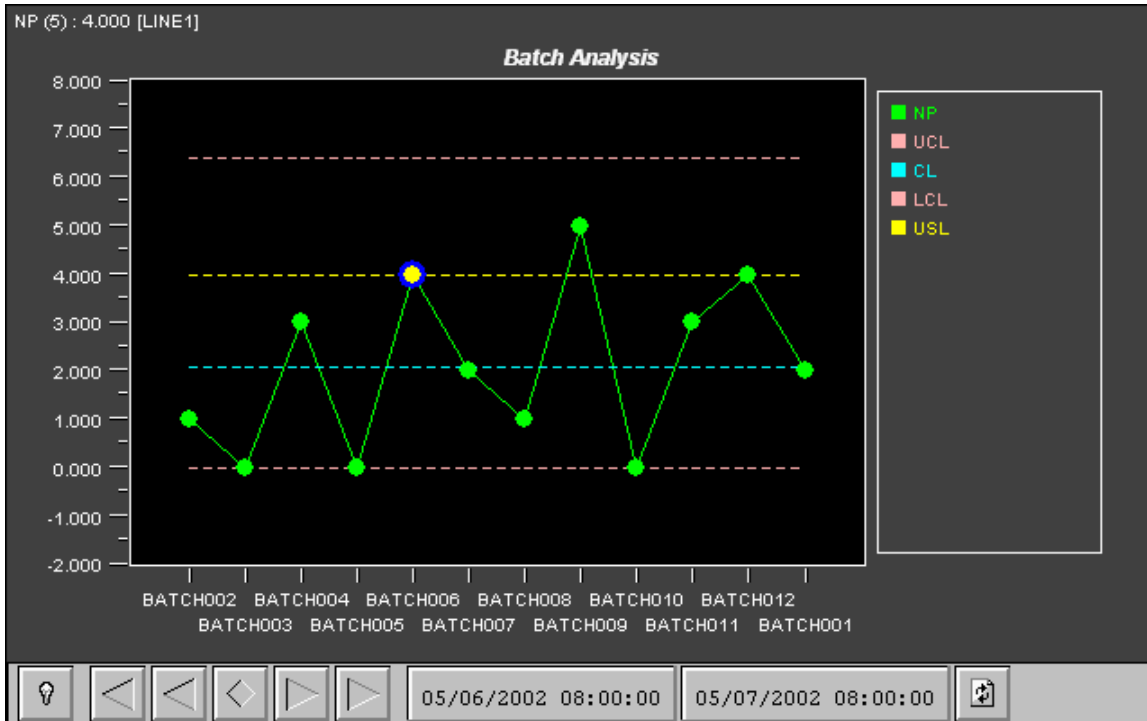
HISTOGRAM



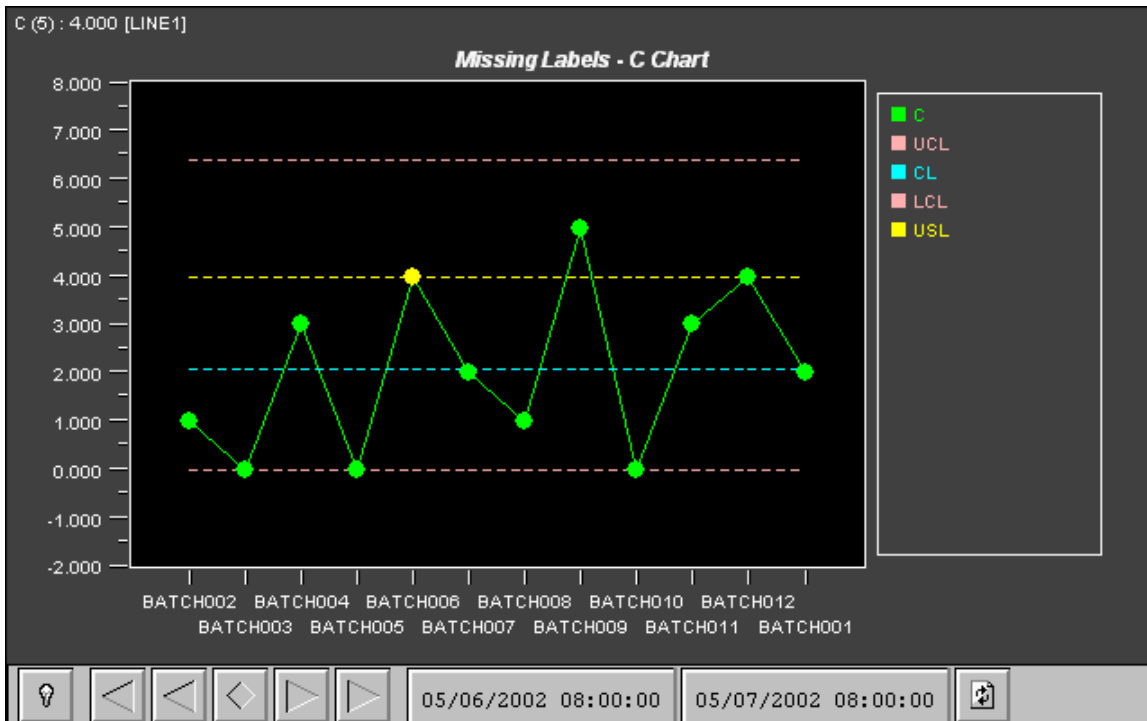
BOX-WHISKER



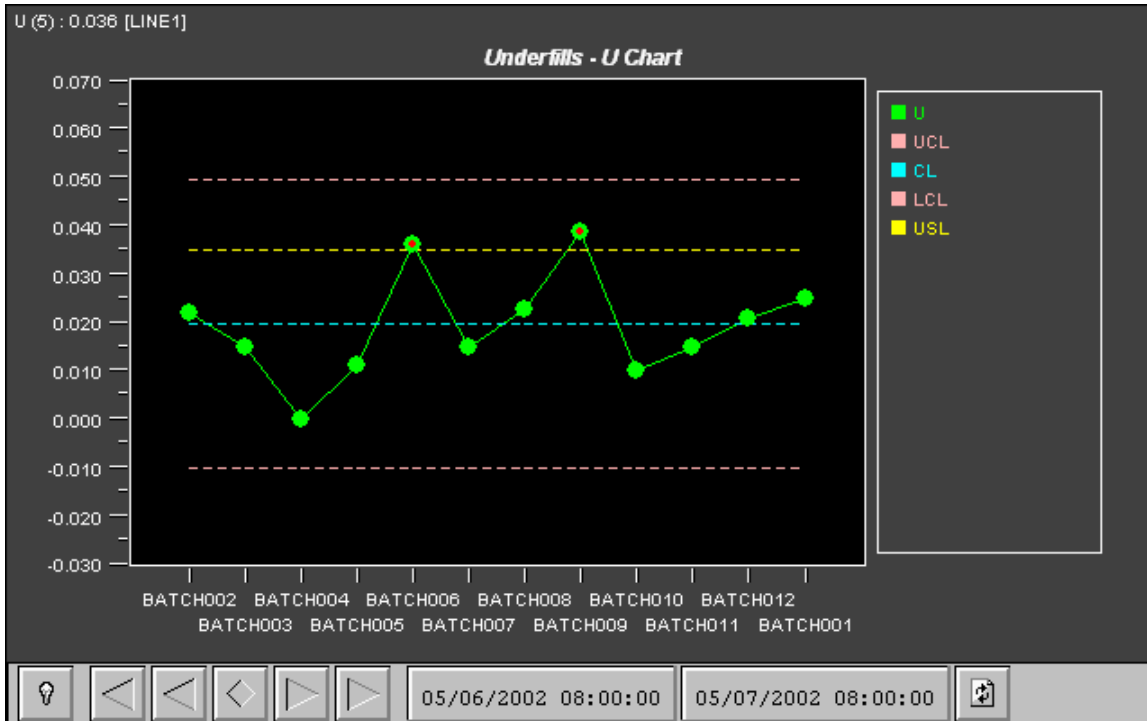
P



NP



C



U

iGrid Applet

WIP Lots

LotID	Location	StartedQty	QueuedQty	RouteName	OperationName	WorkState
BATCH0002029	Compounding	1000.00	0.00	Dispersion	Compounding	IN_PROCESS
BATCH0002030	Compounding	1000.00	0.00	Dispersion	Compounding	IN_PROCESS
BATCH0003001	Compounding	0.00	1000.00	Dispersion	Compounding	QUEUED
BATCH0003002	Compounding	0.00	1000.00	Dispersion	Compounding	QUEUED
BATCH0003003	Compounding	0.00	1000.00	Dispersion	Compounding	QUEUED
BATCH0003004	Compounding	0.00	1000.00	Dispersion	Compounding	QUEUED
CO0002012	Assembly	0.00	1500.00	Cartridge	Make Cartridge	QUEUED

Grid Mode

Current Line Status

Speed	99.80	74.30	15.30	63.50
Yield	88.90	41.20	88.50	34.40

Scoreboard Mode

Machine Status

Machine Name	Maintenance Status	Operational Status	Light Indicators					
Coater71	READY	IN_USE	●	●	●	●	●	●
MillsSystem10	READY	IN_USE	●	●	●	●	●	●
MillsSystem20	READY	IN_USE	●	●	●	●	●	●

MultiLights Mode

	Running	Code 1	Code 2	Code 3	Code 4
Line 1 Filler					
Line 1 Packer					
Line 1 Palletizer					

ColumnLights Mode

03/21/2000 13:29:14	L1Machine1Run	L1Motors	DISC	
03/21/2000 13:29:14	L1Machine3Down1	L1Motors	DISC	
03/21/2000 13:28:36	L2Speed	L2	HIHI	
03/21/2000 13:25:21	L2Yield	L2	LOLO	

Lights Mode




	1st Lot	2nd Lot	3rd Lot	4th Lot	5th Lot	6th Lot
Lot Identifier	BATCH0002001	BATCH0002002	BATCH0002003	BATCH0002004	BATCH0002005	BATCH0002006
Material Code	Dispersion	Dispersion	Dispersion	Dispersion	Dispersion	Dispersion
Location	NONE	NONE	NONE	NONE	NONE	NONE
Qty	1000.00	1000.00	1000.00	1000.00	1000.00	1000.00
Hold	RELEASED	RELEASED	ON_HOLD	ON_HOLD	RELEASED	RELEASED
Quarantine	ACTIVE	ACTIVE	ACTIVE	QUARANTINED	QUARANTINED	ACTIVE

VerticalGrid Mode

DateTime	13:53:19	13:53:29	13:53:39	13:53:50	13:54:00	13:54:10	13:54:21	13:54:31
L1Speed	85.60	88.40	88.70	90.40	86.60	85.80	88.30	89.50
L2Speed	68.80	66.60	61.60	60.80	65.10	65.80	64.60	62.50
L3Speed	9.90	8.40	8.90	10.00	11.40	9.10	12.70	12.00
L4Speed	60.60	64.00	62.00	65.70	66.20	63.00	64.60	68.30

10/26/2000 13:53:19
10/26/2000 13:58:19

VerticalScoreboard Mode

Color	Symbol
Red	
White	
Blue	

EmbeddedLights Mode

iTicker Applet

yr Training Now Offered By HR - Contact Dan Hitz At x26!

Default Mode

iBrowser Applet

- ActiveMetricsQuery
- ActiveMonitorsQuery
- AlertAck
- AlertLogInsert
- AlertLogQuery
- AlertLogSystemQuery
- AlertLogUnackQuery
- AlertUnackInsert
- CategoryDelete
- CategoryInsert
- CategoryListing
- CategoryUpdate
- DatasetDelete
- DatasetDetail

List Mode

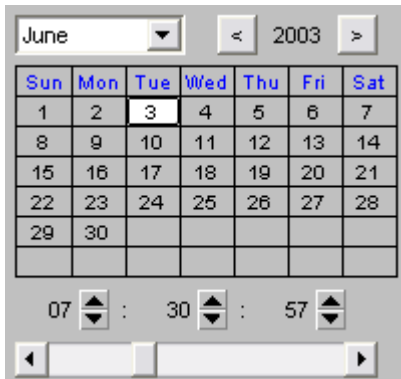
- Line
- Line
- Bar
- GroupBar
- Pie
- Polar
- StackedBar
- FloatingBar
- VariabilityBar
- Gauge
- XY
- XYRegression

Drop-Down List Mode

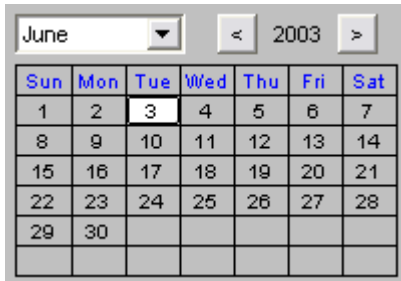


Folder Mode

iCalendar Applet



Date/Time (200 x 188 pixels)

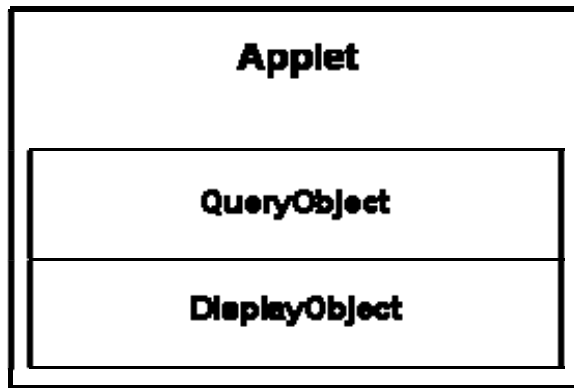


Date Only (200 X 140 pixels)

[Applet Object Model Reference](#)

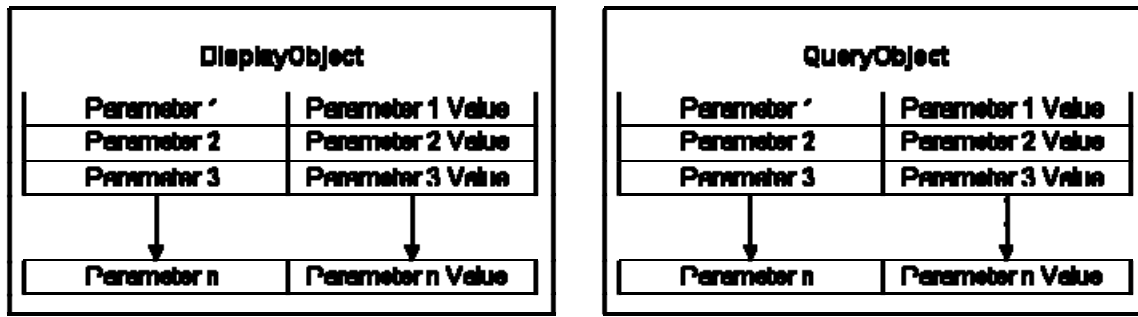
The SAP xApp Manufacturing Integration and Intelligence (SAP xMII) Java applets give the Web site developer a powerful set of interactive tools to help deliver application solutions through a browser. Each applet follows a simple object model which must be understood in order to interact with the applet. The object model is summarized in Figure 1.

Figure 1.



Each applet has a set of basic controls, and has two primary components, a display object and a query object. Each of these objects has a series of parameters that determine their behavior, as shown in Figure 2.

Figure 2.



The display object types are:

- iGridComponent (used by iGrid applet)
- iTickerComponent (used by iTicker applet)
- iChartComponent (used by iChart applet)
- iSPCChartComponent (used by iSPCChart applet)
- iBrowserComponent (used by iBrowser applet)

Similarly, there are a number of query object types:

- TagQuery
- AlarmQuery
- SQLQuery
- XMLQuery
- AggregateQuery
- OLAPQuery

As you may have noticed, the query types match the query template types that may be created in SAP xMII. The display templates correspond to the display template types. Each template has many parameters, but

the parameters have specific meanings depending on the template type. A template is really a grouped set of parameters with specific values. This is an easy way to develop default behaviors and easily reuse settings. At the applet level, the templates are merely paths to get the underlying parameter values. This is why you can override parameter values through applet tags and through script. The parameter values themselves, and not the templates, direct the behaviour of the display and query objects.

When you are attempting to change the behaviour of an applet through Web script, you must know whether the behavior is related to the display object or the query object. This is generally very simple. If the behavior relates to the retrieval of data, you must address the query object. How the applet looks and responds to user input is the display object's job.

To address an applet parameter, you must generally address it at the display object or the query object level. For example, if you want to ask a grid applet which is the selected column, you must first ask the applet to get the display object through the **getGridObject()** command, which will return an `iGridComponent`. You then ask the display object for the selected column number with the **getSelectedColumn()** command.

Similarly, if you wish to change the filter expression for the `SQLQuery` that is displayed in a grid, first ask the grid for the query object with the **getQueryObject()** command. Then use the query object's **setFilterExpr(String)** command to set the new filter value.

The tree below has specific links to reference documents within the SAP xMII online help to help you navigate through the docs and find what you need.

Object Model Detailed Reference

The object model is a hierarchical model. Each object inherits certain capabilities of its parent. You must understand the hierarchy in order to fully take advantage of each object's capabilities. For example, each query object is a child of a `DataQuery` object. The `DataQuery` object has a number of built-in capabilities that each child inherits. Each child (`TagQuery`, `AlarmQuery`, `SQLQuery`, `XMLQuery`) each has its own unique capabilities in addition to the capabilities of `DataQuery`. Therefore, to understand the full exposed functionality of a `TagQuery` object, you must understand first all the capabilities and interfaces of a `DataQuery` object, and also the capabilities and interfaces of a `TagQuery` object.

The tree below contains hyperlinks to each specific class of objects, as well as representing the hierarchy of inheritance. This tree should make it easier to find specific parameters, as well as parameter accessors, for each type of object. For general information on scripting with SAP xMII applets, see the [General Applet Reference](#) and the [Web Scripting with SAP xMII Objects](#) guide. A specific section is available on how to execute applet methods with Web script.

The display objects (`iGridComponent`, `iChartComponent`, and `iTickerComponent`) are shown in a separate branch of the hierarchy, because they do not inherit methods and properties from the applets. However, to address an display object, you must ask the specific applet instance for the object instance it is using, as described in the [Contained Objects](#) section of the [General Applet Reference](#).

Also note that the `iCalendar` object does not inherit any of the functionality of the `SAP xMII Base Applet`.

[DataQuery](#)

[Common Query Parameters](#)

[Common Applet Methods](#)

[SQLQuery](#)

[Parameters](#)

[Parameter Accessors](#)

[TagQuery](#)

Parameters

Parameter Accessors

AlarmQuery

Parameters

Parameter Accessors

XMLQuery

Parameters

Parameter Accessors

AggregateQuery

Parameters

Parameter Accessors

OLAPQuery

Parameters

Parameter Accessors

Base Applet

Common Parameters

Common Parameter Accessors

Common Method Execution

Common Event Support

iGrid (see below for display object (iGridComponent) methods)

iGrid Methods

iGrid Event Support

iChart (see below for display object (iChartComponent) methods)

iChart Methods

iChart Event Support

iTicker (see below for display object (iTickerComponent) methods)

iTicker Methods

iTicker Event Support

iBrowser (see below for display object (iBrowserComponent) methods)

iBrowser Methods

iBrowser Event Support

iSPCChart (see below for display object (iSPCChartComponent) methods)

iSPCChart Methods

iSPCChart Event Support

iCommand

iCommand Reference

iCommand Methods

iCommand Event Support

iCalendar

iCalendar Methods

iCalendar Event Support

Display Objects

iGridComponent

iGridComponent Parameters

iGridComponent Methods

iChartComponent

iChartComponent Parameters

iChartComponent Methods

iTickerComponent

iTickerComponent Parameters

iTickerComponent Methods

iSPCChartComponent

iSPCChartComponent Parameters

iSPCChartComponent Methods

iBrowserObject

iBrowserObject Parameters

iBrowserObject Methods

Quick Start Guide

One of the greatest things about the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) is the ability to quickly setup and administer the SAP xMII Web site remotely. All that is required is a standard Web browser and access privileges to the Web server. The links below show you how to get a working page up and running in just minutes! (For best results, we suggest viewing the pages in the order listed, but feel free to come back to this section any time for a quick refresher.) If authentication is enabled, the default login for the Administrator role on installation is Admin/Admin.

Setting Up a Connection to a Data Source

Creating Query and Display Templates

Building a Sample Page with Microsoft FrontPage

Building a Sample Page with the Page Generator

Web Scripting with SAP xMII Applets for Interactive User Experience

Adding Database Connections

From the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) main menu, select **Data Services | Data Servers**. A list of predefined servers appears on the **Data Servers** page.

Select the type of data source that you want to configure. The data source type is noted in the Description field. For example, "SQL Database."

Click the **Copy** button.

Enter a server name.

Select the **Enabled** check box to make the server active.

Edit the **ServerURL** parameter. Replace *localhost* with the machine name of the data source; however, if the data source is located on the server machine, *localhost* will suffice.

Edit the values in the other parameter fields.

Verify that the **UserName** is correct.

Enter a password.

Note: The system encrypts all passwords to your data sources for added security.

Click the **Save** button. A confirmation popup window appears.

Click the **OK** button in the confirmation popup window.

Click the **Status** button. The **Connection Status** page appears displaying the status of your data connections. A "1" in the **ConnectionsAvailable** column confirms a connection was made. A "0" means there is no connection to your data source. If you cannot establish a connection, confirm your settings for **User**, **Password**, and **URL**. Also verify that the Web server has the necessary security privileges to communicate with the data server. For more information on setting up a server, see SAP xMII System Configuration.

Note: The status page only shows information on JDBC servers in which a connection pool is built. Most tag-based servers such as UDS's do not respond to a status check.

Creating Query and Display Templates

Once you establish a connection to the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) database, the next step is to develop query and display templates. SAP xMII uses a query template to retrieve data from a configured data source and matches it to a display template to present the data to the user. Together, the query and display templates create a simple component that can be inserted into a Web page. The Template Editor is a user-friendly, point and click environment in which to develop both query and display templates. In the following steps, you will construct a simple historical trend from a Microsoft SQL Server database.

Creating a Query Template

From the SAP xMII main menu, select **Data Services | Query Templates**. The **Query Templates** window appears.

Click the **New** button. The **Template Type** list appears.

Select **TagQuery**.

Click the **OK** button. Tabs appear on the **Query Templates** window.

On the **Data Source** tab, select **InSQL** in the **Servers** list.

Select **History** mode.

Modes refer to how you want to query the database, i.e. you may want current, historical, or statistical information.

Every connector has its own set of modes; although each class of connectors (Tag, SQL, or Alarm) has a predefined list of modes that it is expected to support.

Methods refer to additional characteristics for each mode. Many connectors do not use methods. Skip that setting for now (for additional information, refer to the Connector documentation).

Select the **Tag Query Details** tab (Skip the General and Date Range tabs for now, and rely on system defaults. For more information on these tabs, see the Template Editor User's Guide).

The **Available Tags** list displays the valid list of tagnames for the selected server. This is a typical SAP xMII feature called namespace browsing. SAP xMII queries the list of objects relevant to the search you may be performing, directly from the data source you are attempting to query. In this case, SAP xMII asks the InSQL server for a list of tags, so you may select from a list rather than type tagnames from memory. You can select any tag from this list. In this particular case, the tag list is significantly long. You can use the **Tag Selection Mask** field to filter this list. A percent sign (%) is used as a wildcard entry in the filter criteria. A "%Speed" entry in the **Tag Selection Mask** field returns all tagnames that end in "Speed."

Now transfer the tags from the **Available Tags** list to the **Selected Tags** list. You can use the Select All (>>) button to transfer all tags, or you can use the Select One (>) button to transfer selected tags only.

Our query template is complete. If you do not select a date range, it will default to the last 60 minutes. Click the **Test** button to test the query. If your data source is providing data, you should see it returned in the pop-up window.

Once the query template tests correctly, click the **Save As** button to save the template. Choose a name and location pertinent to the query.

Creating a Display Template

Now that the query template is complete, you need to associate it with a display template. Follow the steps below to create a new display template for this example.

Select Visualization Services | Display Templates.

Click the **New** button. The **Template Type** list appears.

Select **iChart**.

Click the **OK** button.

Select **Line** from the **Chart Type** drop-down list. Line charts are used for plotting historical data. The defaults for the other settings are acceptable. Once again, consult the Template Editor User's Guide for advanced documentation.

Click the **Save As** button.

Select a location from the **Folders** list.

Enter a template name.

Click the **OK** button.

Using FrontPage to Generate an Output Page

After you establish a connection to a database and develop query and display templates, you can utilize our Microsoft® FrontPage® add-in. For advanced documentation on the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) Wizard refer to the HTML Add-In User's Guide.

Open FrontPage. Select *File --> New --> Web*. A *Web* is a collection of common Web files managed by FrontPage. If you already have a Web site configured, you can add a new page to it. You must open a *Web* on the SAP xMII Web server to view the query and display templates.

Choose *One Page Web* and select the save location. The save location should be on or below the Web root directory of your Web server. Hit *OK*. By default, a new Web page will open. Now we can use the *SAP xMII Wizard* to insert the HTML code for the applet.

But first we must install it. Go to *Tools --> Add-Ins*.

Browse to C:\Lighthammer\Illuminator\Wizards.

Select the *Illum10FPWizard.dll*. Hit *OK*.

Click on the check-box next to the *SAP xMII Wizard*. Hit *OK*.

Select *Tools --> Illuminator Object*.

The SAP xMII Wizard will load.

Use the windows on the left to browse to the locations of the query template and the display template we created in the Quick Start - Templates section. You can use the buttons on the right to filter query and display types. The bottom section allows you to add parameters to overload the default characteristics of the templates. Use the *Create HTML* button to insert the applet code into our HTML page. (The Add-In is explained in much greater detail in the HTML Add-In User's Guide.)

Now we must save the page into our Web. Hit *File --> Save As*. Give the chart a name and hit *Save*. Remember to save the page to a location at or below your Web server's root directory.

Finally we can load the page using a Web browser. It is important to address the page using the Web server and not the local file system. For the applet to pull data you must use the HTTP protocol.

Using the Page Generator

Open the main menu.

Select Visualization Services | Dynamic Page Generator.

Note: Some experience with HTML is required since you must cut and paste the applet code into a Web page saved on the Web root folder.

Browse in the upper left-hand pane and select the query.

Browse in the lower left-hand pane and select a display template.

Enter dimensions for the applet.

Click the **Generate Page** button. A window appears displaying the applet.

Right-click in the white space and select **View Source**.

Highlight the HTML between and including the <APPLET> tags. This is the code for the JAVA applet displayed on the screen. You can copy the code into an existing Web page. Alternatively, you can save the page on the Web root folder and access it with a Web browser.

Template Editor

Template Editor General Reference

Template Editor

You can use the Template Editor to author query and display templates.

To work with query templates, select Data Services from the main menu and then select Query Templates. The Template Editor for creating and modifying query templates appears.

To work with display templates, select Visualization Services from the main menu and then select Display Templates. The Template Editor for creating and modifying display templates appears.

Using The Template Editor

This section describes the common features of Template Editor applets. The Template Editor is typically accessed via the main menu. For specific template editor functionality, see the appropriate sections of this help chapter. This guide will show you how to configure each type of object through the Template Editor.

For a more detailed reference on each parameter for the different object types, refer to the Connector Parameter Reference (for connector and query object parameters) and the Applet Reference (for applet parameters/properties).

Note: The GUI of the Template Editor has many configurable components, and as a result, it requires a minimum 1024x768 resolution display. Additionally, the system font should be set to a small font size (not large fonts) or the graphic elements will overlap due to inadequate room to fit all of the components on the page.

Button Bar

At the bottom of the editor are a series of buttons and an indicator path/name of the currently loaded template, or "Untitled" if a new template has been initialized.

New Template

Select the New button to display a template type selection dialog. Using the list box, select the type of template you wish to create. A new template will be created using default values with a name of "Untitled". The appropriate configuration tabs are displayed.

Load Template

Select the Load button to display a template selection dialog. The left-hand navigation tree indicates a list of the folders defined below the Templates subdirectory on the server. When you select a folder, a list of previously saved templates appear on the right. If a template is currently loaded into the template editor, the directory and name of the template file will be selected. If no template is loaded into the template editor, no directory will be selected and the template name defaults to Untitled.

You may also delete a template from this dialog by selecting a template and pressing the Delete button. You can create new sub-directories by pressing the New Folder button.

Save/Save As

Selecting the "Save" button will save the currently loaded template, except in the case of a newly created template, in which case it will behave identically to the "Save As" button. The "Save As" button will open a

very similar dialog to the Load dialog, except that the template name is now an editable data entry field. Selecting a pre-existing template from the list and pressing OK will ask for confirmation before overwriting the original template, but exercise general caution when naming and saving templates.

A folder may be "hidden" from view by using an underscore as the first character of the directory/folder name (although templates may still be placed in this directory and referenced in Web pages).

Test Template

If the currently loaded template is a query template, the Test button can be used to open a new browser window and show the results of the query in HTML, XML, or CSV format. Any errors will also be displayed in this new window.

Token

If the currently loaded template is a query template (does not apply to Tag Query or Xacute Query), selecting the Token button will display a dialog. **[Param.1]** (and other available variable expressions) can be copied and pasted (or typed) from the list of available tokens.

The use of tokens allows the user to set specific parameter variables to include in a query that can be given specific values at the time the query is run. Depending on the query object type and the underlying connector, different parameters may be available. Refer to the specific connector documentation for more details.

For example, a quality chart may be displayed on a page with a list of batches contained in a grid. Depending on which batch is chosen, the quality chart is updated showing the data for that batch. A filter expression in an SQL Query Template could be something like the following:

```
BatchHistory.BatchID = '[Param.1]'
```

Then at runtime, you can reference the query object and set the variable to the specific BatchID value through Web scripting. Other examples of tokens include start and end dates which are referenced with [SD] and [ED] respectively. The use of placeholder parameters is discussed in more detail in the IDBC Connector guide.

User Interface Differences

Unlike typical Microsoft Windows applications, Web applications (either HTML pages or Java applets) tend to use a slightly different metaphor for user input to allow usage on a wide range of operating systems and user interface shells. As such, "double clicking" has no effect on list boxes and tree lists. With these types of platform-neutral applications, selecting an item from a list, followed by pressing an "action" button is the more typical UI metaphor.

[Data Request Guide](#)

Query Construction

The SAP xApp Manufacturing Integration and Intelligence (SAP xMII) applets generally provide the richest user interface for interacting with the SAP xMII server. However, there are many situations, such as live reports, queries from limited functionality devices or wireless devices, queries from external systems for data extraction, and other scenarios where simply querying the SAP xMII server directly is the appropriate technique.

URL Format

The query submitted to a Web server is commonly referred to as a URL (Universal Resource Locator). The URL has several parts. The first part is the most common part, which most people who have ever visited a Web site on the Internet are familiar with. This part looks something like <http://www.lighthammer.com>. This part is actually composed of several smaller parts. Suffice it to say that there is a protocol (HTTP) and an

alias to an address (www.lighthammer.com). These parts are required in the address. The alias will refer to the Web server where SAP xMII has been installed. This part of the URL is usually referred to as just the "server name".

The most important part of the URL for this discussion is the rest of the address. An example of a full direct query could look like the following:

```
http://<servername>/Lighthammer/Illuminator?Server=InSQL&Mode=Current&TagName=
SysTimeSec
```

Immediately following the Web server identification is the URL piece that refers to /Lighthammer/Illuminator. This indicates the address is identifying a server side component called Lighthammer/Illuminator. Following the name of the server side component there is a '?'. This separates the target for the query and the parameters that will be provided to the query.

The parameter string is simply a series of name-value pairs separated by the ampersand (&) character. Many queries require more than the one parameter as shown in the above example.

It is important to note that certain special characters with a predefined meaning in HTML/HTTP, when used in an HTML hyperlink or in a URL, need to be provided as their hexadecimal equivalent. This is not an SAP xMII restriction, but rather, an unfortunate HTML and/or browser restriction. Examples include the equals sign, space character, and many others. These hexadecimal symbols must be used in a URL query to make sure that the proper symbol will be encoded. This is especially true if one of these characters exists inside of a parameter value. For example, in a SQL query the parameter "Query" could have a value like the following:

```
Select+%2a+FROM+myTable+WHERE+myID%3d45
```

This Query corresponds to the string :

```
Select * FROM myTable WHERE myID=45
```

Special Characters

Description	Char	URL Usage
Space		%20 or +
Exclamation point	!	%21
Double quotation marks	"	%22
Number sign	#	%23
Dollar sign	\$	%24
Percent sign	%	%25

Ampersand	&	%26
Apostrophe or single quote	'	%27
Open or left parenthesis	(%28
Close or right parenthesis)	%29
Asterisk	*	%2A
Plus sign	+	%2B
Comma	,	%2C
Dash, hyphen, or minus sign	-	%2D
Dot, period, or decimal point	.	%2E
Forward slash or solidus	/	%2F
Colon	:	%3A
semicolon	;	%3B
Less than	<	%3C
Equals	=	%3D
Greater than	>	%3E
Question mark	?	%3F
At sign	@	%40
Open or left square bracket	[%5B

Backslash	\	%5C
Close or right square bracket]	%5D
Caret	^	%5E
Underscore	_	%5F
Acute, back quote, or back tick	`	%60
Open or left curly brace	{	%7B
Vertical bar		%7C
Close or right curly brace	}	%7D
Tilde	~	%7E

Query Parameters

As mentioned above, interacting with SAP xMII is as simple as submitting a Web query via a URL from a Web browser or other system. Web queries are a special form of URL, where the first part indicates the recipient or target application for the query, and the second part contains parameters to be passed in the request. An example shown below queries the current SAP xMII status:

`http://<servername>/Lighthammer/Illuminator?Service=SystemInfo&Mode=Status`

A Parameter can be thought of as a pair of values separated by an equal sign. This is commonly referred to as a name-value pair. The name part being the left side of the equal sign while the value part is the right side of the equal sign. For example, the combination **Mode=Status** means that there is a parameter with the name Mode and this query sets the value to Status. You have probably encountered Web queries if you have ever checked the weather, traded stocks, or bought products on-line using your Web browser. The Service parameter identifies the SAP xMII service that should accept and process this query.

A subset of the connector parameters are supported by all connectors. These include data source specification parameters such as Connector, Server, Mode, and Method, as well as common parameters for date/time specification and date format, output format, and row count. There are also parameters that are common to each "class" of query (Tag, SQL, Alarm, or XML). However, the actual implementation of some of these parameters varies by connector. A summary of the supported parameters is presented in the Connector Parameter Reference guide. This reference details the specifics for each parameter and gives examples of the usage for each.

Parameter Types

There are many types of parameters. The most common types are: String, StringList, Date, Boolean, Integer, and Floating Point. This section will briefly describe the format of these types of parameters.

String Types

String values consist of alphanumeric content. Many string parameters are "free format", others have a wide range of values but require a specific format, and certain parameters expect their value to be provided from a finite list of choices. For example, a "free format" parameter would be the "Title" parameter to the ChartServlet. An example of a parameter that requires a specific format but can take on a wide range of values would be the "Time" parameter to a query, which must be of a specific format such as "today@08:00:00". Lastly, a parameter such as the "DurationUnits" parameters must be provided from the finite list of string constants.

String List Types

String lists are passed as comma-delimited string values. The "Columns" and "Tables" parameters for SQL queries are a good example of this type. This allows several values to be provided within the same parameter name. An example would be:

```
Columns=EmployeeID,FirstName,LastName,SocialSecurityNo
```

Date Types

Date parameters must take the form of a Java date format string. The actual date format can be customized, but any date/time parameters provided must be provided in a format identical to the DateFormat parameter. If the DateFormat is not provided, the default date format is applied, which is MM/dd/yyyy HH:mm:ss. In this case, the date parameters might be "04/08/2000 14:15:13". If the DateFormat parameter was dd-MMM-yyyy HH:mm, the same value would be passed as "08-APR-2000 14:15".

Boolean Types

Boolean parameters are those that are either true or false. All of these parameters can accept one of: true, yes, t, y, or 1 to indicate a "true" value. Likewise they can all accept one of: false, no, f, n, or 0 to signify a "false" value. The ShowLegend parameter is a good example of a boolean data type.

Integer Types

Integer types are simply those parameters that are expecting a whole number as the value. The "RowCount" parameter is an example of this parameter type.

Floating Point Types

Floating point types are those parameters that are expecting a number as the value. This number may include a decimal point and fractional value. The "TotalizerFactor" parameter is an example of this parameter type.

Parameter Priorities

The parameters are processed by the SAP xMII server in a predefined and consistent order of priority. Specific parameter values provided in the query via a URL have the highest priority. Equal in priority to URL entered name-value pairs would be <PARAM> tags embedded in a <SERVLET> tag if using server-side include techniques. Note that these methods of using the system are mutually exclusive and therefore have identical priority. The next highest priority would be the value of parameter defined in an XML query template or display template referenced in the URL. If templates are utilized, the template values are merged with the default values configured in the default template type for each of the supported XML template types (TagQuery, AlarmQuery, SQLQuery, XMLQuery, iChart, iSPCChart, iGrid, iTicker, and iBrowser). The next priority would be the server defaults defined (and customizable) in the system configuration. Finally, there are a number of system defaults that are not modifiable and are built into the software components themselves. A summary of these priorities is presented in the table below.

Server Parameter Priority Hierarchy

Priority	Method	Comment
Highest (1)	URL parameters or servlet/applet parameters	Embedded in the URL or specified in <SERVLET> and <PARAM> tags in an report document, or in the case of Applet usage, specified in <APPLET> and <PARAM> tags in an HTML document
2	Template Files	Parameter values contained in Query or Display XML Templates
3	Defaults in the Default Templates	Defaults provided in the default templates for the query and display templates
4	Defaults in the system configuration	Defaults provided at the Connector level in the system configuration
Lowest (5)	Intrinsic system defaults	Embedded in the software components

All parameters in these three sources can be used in conjunction to form a complete query. It is only in the case when a duplicate parameter name exists in more then one of these sources that the prioritization will take effect.

Query Modes

The Mode parameter defines what type of function the connector is to perform. Every connector has its own set of modes, although each class of connectors (Tag, SQL, Alarm, XML, or Aggregate) has a predefined list of modes that it is expected to support as well. To obtain a list of the supported modes for a given connector a ModeList query can be performed. Such a query could look like the following:

`http://<servername>/Lighthammer/Illuminator?Server=MyServer&Mode=ModeList`

This will produce a list box with all of the supported modes for the connector. To get the list of modes for other servers that may be installed simply replace the MyServer name with the name of the installed server.

Here are some typical modes that will be supported by each class of connector:

Tag Connectors

Current

GroupList

History

HistoryEvent

ModeList

Statistics

TagList
SQL Connectors
ColumnList
Command
FixedQuery
FixedQueryWithOutput
ModeList
Query
TableList
Alarm Connectors
AlarmHistory
AlarmHistoryColumnList
AlarmStatistics
AlarmSummary
AlarmSummaryColumnList
ModeList
XML Connectors
ModeList
XMLQuery
Xacute Connectors
ModeList
Query
TransactionFolderList
TransactionInputList
TransactionList
TransactionOutputList
OLAP Connectors
CubeList
DimensionList
MeasureList
ModeList

Query

Aggregate Connectors

ModeList

Query

SAP xMII System Connector

See the SAP xMII System Connector User's Guide

Note that not all connectors of a specific class will implement all of the recommended modes, and connectors are free to add new modes. Refer to the connector-specific documentation for detailed lists of supported capabilities in each connector.

Namespace Queries vs. Data Queries

Most connectors support both namespace queries and data queries. Namespace queries provide lists of names, as one might expect. This is a key piece of the SAP xMII functionality which enables "discovery" of a connector's functionality and of the data values available from a specific server. When queried directly, these are usually returned in an HTML list box. Combining this type of query with advanced HTML and JavaScripting techniques provides the ability to create an interactive or "ad-hoc" form using the results of the namespace query. Generally, any mode that has a name ending in "LIST" is a namespace query and will return a list box. These same types of queries are used extensively within the SAP xMII Template Editors and with the ad-hoc data browsing dialogs in the SAP xMII Applets. Examples of namespace query modes include ServerList, GroupList, TagList, TableList, ColumnList, ModeList, TimePeriodList, and others.

Data Queries will always return some type of data from the connector. Some connectors may also provide a legend describing the data items returned in the query. Examples of data query modes include History, HistoryEvent, Current, Statistics, Query, FixedQuery, Command, AlarmHistory, AlarmSummary, AlarmStatistics, and others.

[Query Editor General Reference](#)

Query Editor

Queries are the backbone of SAP xApp Manufacturing Integration and Intelligence (SAP xMII). They allow access to a heterogeneous blend of data sources through a simple point and click interface. Once a query is configured, it may be used over and over again on different Web pages and in different visual objects without repeating any definition. In addition, all queries can be modified at runtime to produce different results, or different subsets of the original dataset through the use of query parameters. For a description of the SAP xMII queries, see the Query Construction Guide.

Query templates are configured through the Template Editor. There are a number of common elements between all SAP xMII queries (Tag queries, SQL queries, XML Queries, and Alarm queries). These elements and their configuration are described below. Specific query configuration parameters can be found in the guides for the specific query types.

Data Source Selection

This section is used to choose the source of the data for the query. Only the appropriate connector types are displayed, depending on the type of query to be configured. The server, mode, and method can be selected from this page. Selecting a Server from the list updates the Modes list with individual server specific modes.

All enabled SAP xMII remote servers are also available in this view. Modes are described in the Query Construction Guide.

General Query Parameters

This section is used to assign some common parameters for the query such as the maximum number of rows to be returned from the query as well as the numeric format to be utilized when outputting numeric data values.

The Inline Transform gives the user the ability to assign a stylesheet to manipulate the data further after it has been retrieved through a system connector. An example would be the need to perform calculations to provide summary data. This stylesheet is intended for pre-processing of raw data. Another stylesheet can always be used to format the output of the data through the use of the Stylesheet parameter.

Query Caching

The query caching parameters define whether a query should be cached, and if so, how long the cached data should be considered "fresh." Once a cached data set has been in cache for the entered duration, the next time the same query is called, the data will be retrieved again from the data source. In order for a query request to be cached, it must have the same parameter values, including the same start time and end time.

If a query is a parameterized query, once all the parameters are parsed into name-value pairs, the name-value pairs must match exactly for the query to be retrieved from cache. If the query is set for caching, but the parameters are set at runtime and change, a new cache will be created for each unique set of name-value pairs. Only the most recent instance of a query template will be cached. that means that if at runtime you change the parameter values, the previous cache instance will be overwritten.

For example, if you have a single query defined that brings back all the material usage for a batch from a database, the query may look like the following:

```
Select
b.BatchID,
b.BatchName,
m.MaterialID,
m.MaterialName,
m.Quantity
from
Batch b,
BatchMaterialHistory m
where
b.BatchID = M.BatchID and
b.BatchID = [Param.1]
```

In this example, Param.1 may will be passed in at run time from an HTML page, Agent, or other user interaction. But the query and the result set will be different for each BatchID requested. Only the last instance of the query to be run will be saved into cache.

To clear the cache, use the following URL:

<http://<servername>/Lighthammer/Illuminator?Service=QueryCaching&Mode=ClearCache>

[Click here for more information on Number Formatting.](#)

Date Range Selection

This section is used to assign parameters used in a selection of a date range for time-sensitive queries. In a tag or alarm query, the date filter is applied to the timestamp column. In a SQL query, the date filter is applied to the Date Column that is set on the SQL Query Details tab. In terms of priority, the Schedule is the highest, followed by TimePeriod, then Time, and finally by the StartDate/EndDate parameters. The Duration and DurationUnits parameters are utilized whenever the Time parameter is used -or- whenever one or both of the StartDate/EndDate parameters is not provided. The button with the ellipsis (...) next to each date can be used to pop-up a calendar from which dates/times may be selected. The Clear buttons to the right of the Start and End Dates set their respective parameters to blank, and the Clear buttons next to the Time Period

and Schedule lists deselect any selected Time Period or Schedule. The Interval Count is an integer parameter that can be used to divide the overall query time into evenly spaced segments.

When you wish to use a schedule, you should select either the CurrentShift or PreviousShift time period. The default is CurrentShift. Based on the configuration of the Schedule, the current shift or previous shift is dynamically determined by the system and the appropriate time period parameters are used in the query. See Schedules for further explanation of the use of the Schedule Parameter.

The Allow Future Dates check box sets/resets the AllowFuture query parameter. If you have an end date that resolves into the future, you can choose to "truncate" the end date to the current date-time of the server, or use the full time scale. By truncating to now, you can look at a query such as Shift to Date, or Week to Date, etc., but have the end date not go to the full duration. That makes things like charts use the full allocated space for data. If you don't have data that runs into the future, but don't truncate the end date, the chart object may only be half-filled - because the chart X-Axis will be scaled for the full time duration, even if there is no data.

[Click here for more information on Date Formats.](#)

Parameters

This page is used to preload [Param.1] through [Param.32] with a desired value for the query being constructed, if that query uses the token placeholders. For details on the query parameters configured by this page, refer to the Query Parameter Reference Guide and the Query Construction Guide.

When parameters are entered here and the query is saved, they also act as default values, and if no values are provided at runtime, the values entered here will be used.

Transform

This page is used to define an inline transform and to define the parameters to be used by the transform. For more information on inline transforms, see the In-Line Transforms guide and the Pre-Defined InLine Transform Reference guide. When parameters are entered here and the query is saved, they also act as default values, and if no values are provided at runtime, the values entered here will be used.

Security

Individual templates can be secured for read (use) and write (modify) permissions based on roles. Each function can be assigned to one or more roles. The default reader role is "Everyone" and the default writer roles are "Administrators" and "Developers." Only those roles with write permissions can edit the template. The read permission gives the selected roles the ability to use the template on a Web page. If the user does not have read permission to the template, an applet that uses the template on a Web page does not render data for that user. This permission capability allows groups to protect their templates from modification by users in other project development groups.

[Tag Query Editor Reference](#)

TagQuery Editor

This editor is used to create templates that will access Tag-oriented connectors, such as process historians and Human Machine Interface (HMI) software applications, which typically provide current value, historical (time-series) data values, and statistical summaries of historical data.

Data Source Selection

This section is used to choose the source of the data for the query. Only the appropriate Tag servers are displayed. The server, mode, and method can be selected from this page. For Tag queries, the method drop down combo contains a list of general statistical methods. Refer to the connector-specific documentation for any special implications of this value, or additional methods that are not included in the

default list. Methods not included in the default list, or different combinations of methods may be manually entered in the method data entry box.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab.

Tag Query Details

This page is enabled whenever the mode is set to any available mode other than "ModeList". This page is used to select the tags that will be displayed in a query, as well as to specify data retrieval parameters that affect the behavior of history and statistical queries.

For servers that support tag groups, the Available Groups list box displays all groups of tags in a tree-like format. Selecting a tag group name populates the Available Tags list box with all of the individual tag names contained in the selected tag group. For servers that do not support tag groups, the Available Tags list box displays all available tags. The Apply button filters the tags shown in the Available Tags list box with the tag name wildcard pattern entered in the Tag Selection Mask data entry box. If no matching tag names are found, the list displays no tags. The Clear button removes any entered tag selection mask and refreshes the Available Tags list box.

Tags can be selected in two primary ways as follows:

The first technique is to specify each tag name by adding it to the Selected Tags list box on the right side of this page. The Add All (>>) button populates the Selected Tags list box with up to the first 128 tags displayed in the Available Tags list box. The Add Selected (>) button populates the Selected Tags list box with up to the first 128 selected tags displayed in the Available Tags list box. When checked, the Allow Multiple Selections checkbox configures the Available Tags list box to select or unselect an individual tag for each mouse click. Unchecking the checkbox will remove the selection highlight from all selected tags in the list. The Selected Tags list box prevents multiple instances of any individual tag name. The Remove All (<<) button removes all tag names from the Selected Tags list box, and the Remove Selected (<) button removes a single selected tag name from the list box. The Up and Down buttons move a selected tag name in the Selected Tags list box up one row and down one row respectively. Tags can also be added to the Selected Tags list box by manually entering the tag name into the Tag Name input box and pressing the associated Add button, thus allowing for off-line development.

The second technique is to use the Group and Mask mechanism to ask SAP xMII to do a "wildcard" (%) match of tag names matching a particular pattern and, optionally, contained in a specified group. The mask option is also a convenient way to filter large tag databases when browsing tags in this page. When checked, the Use Group/Mask For Selection checkbox saves the entered Mask and/or selected Group with the template, and ignores any tags shown in the Selected Tags list box. When clear, the Use Group/Mask For Selection checkbox saves only tags shown in the Selected Tags list box and ignores any group or mask values.

In general the Resolution parameter is used to provide retrieval resolution for interpolated values when performing a "History" query, and the TotalizerFactor parameter is used as a conversion factor for the TOT method when performing a "Statistics" query. The behavior of the Resolution and TotalizerFactor parameters are connector-specific. Refer to the connector documentation for specific support and usage of these parameters for each connector.

After selecting the desired tag names or group/mask properties, the Test button can be used to preview the results of the query.

Values

Certain tag servers have the ability to accept tag write requests. The Values tab lists the tags that are part of the write query and allows you to enter test/default values that are to be written for each tag.

Parameters

Refer to the Query Editor General Reference for more information on this tab.

Transform

Refer to the Query Editor General Reference for more information on this tab.

[SQL Query Editor Reference](#)

SQLQuery Editor

This editor is used to create templates that will access SQL-oriented connectors, such as the ODBC connector. This connector is used to access a wide range of database servers. Because of the flexibility of the ODBC connector, a number of tools have been provided to assist in the development and testing of queries.

Data Source Selection

This section is used to choose the source of the data for the query. Only the appropriate SQL connector types are displayed. For Alarm queries, the method drop down combo contains a list of general statistical methods. Refer to the connector-specific documentation for any special implications of this value, or additional methods that are not included in the default list. Methods not included in the default list may be manually entered in the method data entry box.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab.

SQL Query Details

This page is enabled whenever the mode is set to "Query". The user interface provides a means for selecting tables and columns and for entering selection, filtering, ordering, and grouping criteria. Additionally, there are a series of "Paste" buttons that can be used to insert the database table name and column as selected in the leftmost "Available" list boxes to the associated expression builders on the right side of the page. The Paste buttons use the 'Table.Column' format, thus requiring that both a table and column name must be selected in the two Available list boxes. The Join, Filter, Sort, and Grouping expressions can also be entered manually.

The functionality of the various buttons located between both pairs of Available and Selected list boxes is very similar for Tables and Columns, so unless specified, the following operational discussion is common to both pairs of list boxes: The Add All (>>) button populates the Selected Columns list box with all of the corresponding Available Columns list box. The Add Selected (>) button populates the Selected list box with all of the selected items in the Available items list box. The Remove All (<<) button removes all items from the Selected list box, and the Remove Selected (<) button removes a single selected item from the Selected list box. The Up and Down buttons move a selected item in the Selected list box up one row and down one

row respectively. When checked, the Allow Multiple Selections checkbox configures the Available Columns list box to select or unselect an individual column for each mouse click. Unchecking the checkbox will remove the selection highlight from all selected columns in the list. Items can also be added to the Selected list box by manually entering the table or column name into the appropriate input box and pressing the associated Add button, thus allowing for off-line development.

The DateColumn parameter can be entered manually or assigned to the currently selected column in the Available Columns list box by pressing the Set button. This parameter, when provided, will enable SAP xApp Manufacturing Integration and Intelligence (SAP xMII) to automatically append the appropriate SQL syntax to specify a date range as configured on the Date Range page previously discussed.

After editing the desired items (a minimum of one table and one column are required), the Test button can be used to preview the results of the query.

Fixed Query Details

This page is enabled whenever the mode is set to Command, FixedQuery, or FixedQueryWithOutput. You use FixedQuery mode when you want to enter straight, native SQL code or to execute stored procedures in the data source. The user interface provides a means for selecting tables and columns and for entering the SQL query text and query parameter values. Additionally, there are two "Paste" buttons that can be used to append the name of a database table or column as selected in the Available Tables and Available Columns list boxes to the Query expression text box on the right side of the page. The Query and Query Parameters expressions can also be entered manually, the Query Parameters expression being entirely manual. Query Parameters are typically used to enter input parameters for stored procedures. Refer to the Parameter Reference for SQL section for more information. To the right of the Query expression builder text box is a collection of buttons which provide query syntax building tools. After editing the appropriate items, the Test button can be used to preview the results of the query.

Parameters

Refer to the Query Editor General Reference for more information on this tab.

Transform

Refer to the Query Editor General Reference for more information on this tab.

[Alarm Query Editor Reference](#)

AlarmQuery Editor

The AlarmQuery Editor creates templates that access alarm-oriented connectors. These connectors access alarm and event management systems such as those provided with DCS systems, HMI/SCADA packages, building automation systems, and network management/monitoring systems.

The tabs on the AlarmQuery Template Editor are explained below.

Data Source

You select the source of the data for the query in the Data Source tab. Only the appropriate alarm servers are displayed. For alarm queries, the Method drop-down list box contains a list of general statistical methods. Refer to the connector-specific documentation for any special implications of this value or additional methods that are not included in the default list. Methods not included in the default list may be entered manually in the Method field.

See the Query Editor General Reference topic for more information on this tab.

General

See the Query Editor General Reference topic for information on this tab.

Date Range

See the Query Editor General Reference topic for information on this tab.

Alarm Query Details

The Alarm Query Details tab is enabled when the mode is set to any type of data query (rather than a namespace query). You can select column names that can be used in construction of the optional FilterExpr and SortExpr parameter values. The FilterExpr and SortExpr parameters can be thought of as a SQL WHERE clause and ORDER BY clause expression respectively.

The Paste buttons can be used to append the selected alarm/event column to any of the expression builders on the right. The filter and sort expressions can be entered manually. The Available Columns list depends on the query mode and method. The buttons to the right of the Filter Expression and Sort Expression boxes enable syntax building. You can use the Test button to preview the results of the query.

Parameters

See the Query Editor General Reference topic for information on this tab.

Transform

See the Query Editor General Reference topic for information on this tab.

Security

See the Query Editor General Reference topic for information on this tab.

[XML Query Editor Reference](#)

XMLQuery Editor

This editor is used to create templates that will access external XML data in formats such as static XML documents, URL queries, or dynamically generated XML from the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) Agent, and return the data in standard SAP xMII format for XSL transformation or any other standard visualization tools. The XMLConnector provides a powerful means for integrating SAP xMII with enterprise data.

Data Source Selection

A single predefined server alias (XML Connector) is used for any and all queries. This section is used to choose the source of the data for the query. Only the appropriate XML servers are displayed. The server, mode, and method can be selected from this page.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab.

XML Query Details

This page is enabled whenever the mode is set to "XMLQuery". The user interface provides manual input boxes for entering the Source and Transform URLs. Please refer to the Parameters Specific To XML Connectors section of the Query Parameter Reference Guide for details on the requirements for the Source

URL and Transform URL values. After editing the appropriate items, the Test button can be used to preview the results of the query.

Parameters

Refer to the Query Editor General Reference for more information on this tab.

Transform

Refer to the Query Editor General Reference for more information on this tab.

[Aggregate Query Editor Reference](#)

AggregateQuery Editor

The AggregateQuery Editor is used to create templates that combine the results of multiple (up to 32) SAP xApp Manufacturing Integration and Intelligence (SAP xMII) query templates into a single result set. That result set can then be used just like any other SAP xMII query.

Data Source

The Data Source tab is used to choose the source of the data for the query. When creating an aggregate query, you must choose the Aggregate Connector as the server.

Refer to the Query Editor General Reference for more information on this tab.

Template Selection

This section is used to choose the individual templates that are combined to create the single result set of the aggregate query. You can browse the server templates and add/remove the selections one at a time using the arrow keys. When you select a template, you must give it an alias name. This allows greater flexibility when creating template parameter mappings.

General

Refer to the Query Editor General Reference for more information on the General tab.

Date Range

Refer to the Query Editor General Reference for more information on the Date Range tab.

Aggregate Query Details

The Query Details tab contains critical configuration steps for the aggregate query. In order to pass parameters to the underlying queries, you must explicitly map the aggregate query parameters to the underlying query template parameters. There are two ways to map parameters to the member queries.

First, you can map a parameter to the same parameter for all the member queries at once. Second, you can map any parameter to a parameter in a single member query.

A convenient feature is available through the Set Common Target Params button. It sets commonly mapped parameters including all of the time parameters for time sensitive queries. If you are using a time sensitive query and you want all of the underlying queries to use the same time and duration, you should map all of the time parameters to all of the underlying queries.

Refer to the Parameters Specific To Aggregate Queries section of the Query Parameter Reference Guide for details on the requirements for the query and target mapping parameters.

Parameters

Refer to the Query Editor General Reference for more information on this tab.

Transform

Refer to the Query Editor General Reference for more information on this tab.

[Xacute Query Editor Reference](#)

Xacute Query Editor Reference

This editor is used to create query templates that take the output of an SAP xApp Manufacturing Integration and Intelligence (SAP xMII) Business Logic Services transaction and send the results to an SAP xMII applet. The query template result set can then be used just like any other SAP xMII query.

Data Source Selection

This section is used to choose the source of the data for the query. When creating an XacuteQuery, you must choose the XacuteConnector as the server.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab. In order to use the built-in data handling functions within an SAP xMII query, you must map in a start and end date parameter on the Input Params tab, described below.

Transaction

The Transaction tab is where you define which SAP xMII Business Logic Services transaction you wish to run when the query is executed. You pick the folder and a specific transaction. When you select a transaction, all transaction properties marked as output parameters are listed in the Outputs list box. you can select a specific property to be output, or you can type in "*" to receive all the scalar outputs (non-XML data types) available from the transaction. If you wish to receive an XML document as the output, you can must specify the single output parameter that represents the XML document in the transaction.

Please refer to the Parameters Specific To XacuteConnectors section of the Xacute Connector Guide for details on the requirements for the query and mapping parameters.

Input Parameters

Let's assume you have an transaction properties for start date, end date, and lot ID named LotStartDate, LotEndDate, and MaterialLotID, respectively. If you wish to dynamically set these values via the query template, you would need to use a combination of mapped parameters and mapped date parameters. This allows you to name the transaction properties however you wish. The SAP xMII query parameters must be Param.1 through Param.32. In order to link the user defined SAP xMII Business Logic Services transaction property names to SAP xMII query parameters, you must use mapped parameters.

In the above example, you would set MappedParamTarget.1 = MaterialLotID. You could also specify a value in order to test the query. You would specify LotStartDate as the StartDateMappedParam and LotEndDate as the EndDateMappedParam. By mapping the start and end dates, you can use all the built-in applet date handling functions, including the applet "VCR" buttons.

In the example below, we are mapping the SAP xMII Business Logic Services Property "AirportCode" to Param.1, and setting a default value of PHL in order to test the query.

Transform

Refer to the Query Editor General Reference for more information on this tab.

Xacute Query Editor Reference

This editor is used to create query templates that take the output of an SAP xApp Manufacturing Integration and Intelligence (SAP xMII) Business Logic Services transaction and send the results to an SAP xMII applet. The query template result set can then be used just like any other SAP xMII query.

Data Source Selection

This section is used to choose the source of the data for the query. When creating an XacuteQuery, you must choose the XacuteConnector as the server.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab. In order to use the built-in data handling functions within an SAP xMII query, you must map in a start and end date parameter on the Input Params tab, described below.

Transaction

The Transaction tab is where you define which SAP xMII Business Logic Services transaction you wish to run when the query is executed. You pick the folder and a specific transaction. When you select a transaction, all transaction properties marked as output parameters are listed in the Outputs list box. you can select a specific property to be output, or you can type in "*" to receive all the scalar outputs (non-XML data types) available from the transaction. If you wish to receive an XML document as the output, you can must specify the single output parameter that represents the XML document in the transaction.

Please refer to the Parameters Specific To XacuteConnectors section of the Xacute Connector Guide for details on the requirements for the query and mapping parameters.

Input Parameters

Let's assume you have an transaction properties for start date, end date, and lot ID named LotStartDate, LotEndDate, and MaterialLotID, respectively. If you wish to dynamically set these values via the query template, you would need to use a combination of mapped parameters and mapped date parameters. This allows you to name the transaction properties however you wish. The SAP xMII query parameters must be Param.1 through Param.32. In order to link the user defined SAP xMII Business Logic Services transaction property names to SAP xMII query parameters, you must use mapped parameters.

In the above example, you would set MappedParamTarget.1 = MaterialLotID. You could also specify a value in order to test the query. You would specify LotStartDate as the StartDateMappedParam and LotEndDate as the EndDateMappedParam. By mapping the start and end dates, you can use all the built-in applet date handling functions, including the applet "VCR" buttons.

In the example below, we are mapping the SAP xMII Business Logic Services Property "AirportCode" to Param.1, and setting a default value of PHL in order to test the query.

Transform

Refer to the Query Editor General Reference for more information on this tab.

OLAP Query Editor

This editor is used to create query templates that take the output of an OLAP datasource MDX query and send the results to SAP xApp Manufacturing Integration and Intelligence (SAP xMII). The query template result set can then be used just like any other SAP xMII query. See the OLAP Connector reference for additional information.

Data Source Selection

This section is used to choose the source of the data for the query.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab.

OLAP Query Details

This tab uses namespace browsing to help the user create an MDX query expression. You must be familiar with MDX syntax in order to create an OLAP query. The Show Null Values As Zero checkbox will set the NullAsZero parameter. If set to true, null numeric values are replaced with zeros. If false, nulls are replaced with the minimum value for the data type in the Java language (for instance, the minimum value for a Double is 4.9e-324)

Aliases

Aliases are used similarly to the "as" statement in ANSI SQL. If values are supplied on this tab, the returned column names are replaced, in order, by the Alias columns 1 through 128. For example, the column name "Column 1" becomes "Alias1."

Parameters

Refer to the Query Editor General Reference for more information on this tab.

Transform

Refer to the Query Editor General Reference for more information on this tab.

OLAP Query Editor

This editor is used to create query templates that take the output of an OLAP datasource MDX query and send the results to SAP xApp Manufacturing Integration and Intelligence (SAP xMII). The query template result set can then be used just like any other SAP xMII query. See the OLAP Connector reference for additional information.

Data Source Selection

This section is used to choose the source of the data for the query.

Refer to the Query Editor General Reference for more information on this tab.

General Query Parameters

Refer to the Query Editor General Reference for more information on this tab.

Date Range Selection

Refer to the Query Editor General Reference for more information on this tab.

OLAP Query Details

This tab uses namespace browsing to help the user create an MDX query expression. You must be familiar with MDX syntax in order to create an OLAP query. The Show Null Values As Zero checkbox will set the NullAsZero parameter. If set to true, null numeric values are replaced with zeros. If false, nulls are replaced with the minimum value for the data type in the Java language (for instance, the minimum value for a Double is 4.9e-324)

Aliases

Aliases are used similarly to the "as" statement in ANSI SQL. If values are supplied on this tab, the returned column names are replaced, in order, by the Alias columns 1 through 128. For example, the column name "Column 1" becomes "Alias1."

Parameters

Refer to the Query Editor General Reference for more information on this tab.

Transform

Refer to the Query Editor General Reference for more information on this tab.

[iChart Editor Reference](#)

iChart Editor

This editor is used to configure the multitude of properties for the iChart object. The iChart object is a highly flexible and configurable Java applet, and as a result, there are a large number of configurable options and behaviors.

General

This tab is used to configure general chart parameters and behaviors. It is used to select the chart type and overall chart colors and fonts. It is also used to configure which elements of the chart will be displayed. For details on the individual parameters configured by this page, refer to the iChartComponent Properties tables.

Data Mapping

This tab is used to map columns returned by a query to the various elements of the iChart object, notably which column values to chart, which to use when constructing virtual "tag names", which to use when constructing virtual "tag descriptions", which to use when constructing data point labels, and other data from the query that is to be associated with a "tag name". For details on the individual parameters configured by this tab, refer to the iChartComponent Data Binding Properties table.

The right side of the screen can be used to browse to a specific template and then use the appropriate arrow keys to insert column mappings directly into the column attributes on the left side of the screen. This allows an easy way to map specific query values and attributes to specific chart behaviors. It can also prevent typographical errors.

Legend

This tab is used to configure the appearance and behavior of the chart legend which can appear to the right or the top of the chart. For details on the individual parameters configured by this tab, refer to the iChartComponent Legend Properties table.

X Axis

This tab is used to configure parameters associated with the X axis (horizontal axis). For details on the individual parameters configured by this tab, refer to the iChartComponent X Axis Properties table.

Y Axis

This tab is used to configure parameters associated with the Y axis (vertical axis). For details on the individual parameters configured by this tab, refer to the iChartComponent Y Axis Properties table.

Pen Details

This tab is used to configure parameters associated with pen colors and data value scaling. Note that global behaviors will always take precedence over individual pen-specific behaviors. For details on the individual parameters configured by this tab, refer to the iChartComponent Global Scaling And Numeric Precision Properties table.

Behaviors

This tab is used to configure which functions will be enabled or disabled when the user right-clicks the mouse on an iChart object at runtime. For details on the individual parameters configured by this tab, refer to the iChartComponent Behavior Properties table.

Refresh Page

This tab is used to control the automatic refresh behavior for the object. For details on the individual parameters configured by this tab, refer to the General Applet Common Properties/Methods table.

Security

For details on the individual parameters configured on this tab, see the general applet Common Properties/Methods table. See the Security information in the Query Editor topic.

[iSPCChart Editor Reference](#)

iSPCChart Editor

This editor is used to configure the multitude of properties for the iSPCChart object. The iSPCChart object is a highly flexible and configurable Java applet, and as a result, there are a large number of configurable options and behaviors.

General

This tab is used to configure general chart parameters and behaviors. It is used to select the chart type and overall chart colors and fonts. It is also used to configure which elements of the chart will be displayed. For details on the individual parameters configured by this page, refer to the iSPCChartComponent General Parameters tables.

Data Mapping

As discussed in the parameter reference above, using the iSPCChart object requires some data mapping to give the iSPCChart object "hints" as to how to link the columns returned from the query with the various elements in the chart (data values, unique id, attributes, etc.). As an example, suppose that a SQL query returns the following data set:

SaleAmount	OrderID	CompanyName	ShippedDate
2556.95	10393	Save-a-lot Markets	01/03/1997 00:00:00
2505.60	10398	Save-a-lot Markets	01/09/1997 00:00:00
3868.60	10401	Rattlesnake Canyon Grocery	01/10/1997 00:00:00
2713.50	10402	Ernst Handel	01/10/1997 00:00:00
3063.00	10400	Eastern Connection	01/16/1997 00:00:00
9194.56	10424	Mère Paillarde	01/27/1997 00:00:00
11188.40	10417	Simons bistro	01/28/1997 00:00:00
4899.20	10430	Ernst Handel	02/03/1997 00:00:00
4924.14	10440	Save-a-lot Markets	02/28/1997 00:00:00
2684.00	10455	Wartian Herkku	03/03/1997 00:00:00
3891.00	10458	Suprêmes délices	03/04/1997 00:00:00
3849.66	10451	QUICK-Stop	03/12/1997 00:00:00
2518.00	10465	Vaffeljernet	03/14/1997 00:00:00

10495.60	10479	Rattlesnake Canyon Grocery	03/21/1997 00:00:00
3163.20	10490	HILARION-Abastos	04/03/1997 00:00:00
2550.00	10511	Bon app'	04/21/1997 00:00:00
4707.54	10510	Save-a-lot Markets	04/28/1997 00:00:00
4150.05	10518	Tortuga Restaurante	05/05/1997 00:00:00
3192.65	10524	Berglunds snabbköp	05/07/1997 00:00:00
4180.00	10530	Piccolo und mehr	05/12/1997 00:00:00
8623.45	10514	Ernst Handel	05/16/1997 00:00:00
9921.30	10515	QUICK-Stop	05/23/1997 00:00:00
2812.00	10546	Victuailles en stock	05/27/1997 00:00:00
3554.28	10549	QUICK-Stop	05/30/1997 00:00:00
2944.40	10555	Save-a-lot Markets	06/04/1997 00:00:00
2844.50	10561	Folk och få HB	06/09/1997 00:00:00
10191.70	10540	QUICK-Stop	06/13/1997 00:00:00

2519.00	10567	Hungry Owl All-Night Grocers	06/17/1997 00:00:00
3120.00	10588	QUICK-Stop	07/10/1997 00:00:00
4725.00	10595	Ernst Handel	07/14/1997 00:00:00
6475.40	10607	Save-a-lot Markets	07/25/1997 00:00:00
4109.70	10605	Mère Paillarde	07/29/1997 00:00:00
6375.00	10612	Save-a-lot Markets	08/01/1997 00:00:00
4807.01	10616	Great Lakes Food Market	08/05/1997 00:00:00
2697.50	10618	Mère Paillarde	08/08/1997 00:00:00
5510.59	10633	Ernst Handel	08/18/1997 00:00:00
2775.05	10629	Godos Cocina Típica	08/20/1997 00:00:00
4985.50	10634	Folies gourmandes	08/21/1997 00:00:00
2761.94	10637	Queen Cozinha	08/26/1997 00:00:00
2720.05	10638	LINO-Delicateses	09/01/1997 00:00:00

First we want to link this chart to a specific query. Use the browse button (...) to the right of the Query Template Name text box to select an existing query. The query column names will appear in the Column Names list box once a query template is selected.

The value column that we wish to evaluate using statistical analysis is the SaleAmount column. We will use that column as the Value column for the query. Highlight the SaleAmount column in the Column Names list and press the < button next to the Value Columns List. Most SPC charts can have only a single value column.

In general, the < button adds selected columns to the appropriate list, the > button removes highlighted columns from the appropriate list, and you can change the order of the columns by using the up (Up) and down (Dn) buttons.

Suppose we are also interested in seeing other columns while analyzing the chart. Other data returned in the query can add context to the analysis, even though they are not part of the statistical calculations. For instance, it may be interesting to see the company name that is purchasing the product while we are analyzing our sales figures. We will add the OrderID, CompanyName, and ShippedDate as Attribute columns. When you are viewing the chart, you can see the attribute values for each point. They are available in a mode-less window through the right-click menu at runtime. (You may also use the mouseover parameter to see data as well - see below). Highlight the columns that you wish to see, and add them to the Attribute Columns list with the < button. You can order the attributes by using the up (Up) and down (Dn) buttons.

The Sample Size Column is used for attribute charts. If this column is returned via the data query, and you wish to use it instead of a static value (see Parameters), this column should be identified here.

The Time Stamp column is used to identify the date-time column for time sensitive queries. This will also be used to identify "break points" in the control limits - if you change control limits, and give them an effective date, this is the column that will be used to identify which limits to use. This is required for Limit Transactions in the chart.

Documenting SPC Data Points

In order to store comments or to document a specific subgroup point within the SPC chart, you must create a unique identifier for the point or subgroup so that if the same point is retrieved again, the documentation previously entered is available to the chart user. To uniquely identify a point in this query, we will use the OrderID column, since that by definition in our datasource is unique. In a tag datasource, using a HistoryEvent type query, you may need to use both the tag name and a timestamp column from the returning data.

Refer to the iSPCChartComponent Data Mapping Parameters table for more information.

Extended Data Mapping

The extended data mapping configuration tab is for additional, optional mapping parameters. If you return the control and/or specification limits in the data query, you can identify them here. This will override the queries identified on the Limits tab. This can improve performance, because only one query is required to return both the data to be analyzed and the control and/or the specification limits.

The X-Axis label attribute will default to 1, 2, 3,...n. If you add a column to this, the values in the returning columns will be used. In this example, we are using the OrderID as the X-Axis label.

The Highlight Attribute Name is the default highlight attribute column that you wish to use for the chart. When points are selected, those sharing the same value for this column will be highlighted. This may be changed at runtime by the user. The Highlight Attribute Value field will automatically highlight the points having the specified value in the Highlight Attribute Name column when the chart is refreshed, without the user needing to select a point.

The Mouseover Attribute identifies the column of data that will be shown in the upper left-hand corner of the chart when the mouse cursor is passed over a point (must be referenced in the Attribute Columns list). This may also be set at runtime through Web page scripting. For more information on setting applet parameters through script, see the Applet General Reference and the Applet Object Model sections in the help docs.

Refer to the iSPCChartComponent Data Mapping Parameters table for more information.

Parameters

This tab is used to configure the parameters that are specific to individual chart types. For details on each chart type's specific parameters, refer to the relevant table in the Chart Specific Parameters table in the iSPCChart Reference.

Legend

This tab is used to configure the appearance and behavior of the chart legend. For details on the individual parameters configured by this tab, refer to the iSPCChartComponent Legend Properties table.

SPC Axes

This tab is used to configure parameters associated with the X axis (horizontal axis) and Y axis (vertical axis). For details on the individual parameters configured by this tab, refer to the iSPCChartComponent Axis Properties table.

Upper Chart

This tab is used to configure properties of the upper chart. See the iSPCChartComponent Upper Chart Parameters table for more information.

Lower Chart

The Lower Chart tab is virtually identical to the Upper Chart tab, except that the property changes affect the lower chart. These properties are only used if a lower chart is appropriate for the chosen SPC Chart Type such as the Moving Range (MR) portion of an XBAR-MR chart type.

Refer to the iSPCChartComponent Lower Chart Parameters table for more information.

Limits

There are multiple ways to set both control limits and specification limits for a chart:

Specifying a query that returns the control limits and/or the specification limits from another data source (see UpperControlLimitSource, LowerControlLimitSource, and Upper Spec Source in the iSPC Chart Reference).

Directly entering the control limit and/or the specification limit values into the template (see Lower Control Limit, Centerline, Upper Control Limit, Lower Spec Limit, Target, and Upper Spec Limit in the iSPC Chart Reference).

Calculating the control limits on the fly based on the dataset (see the Calculate Control Limits in the iSPC Chart Reference).

You may also return the limits in the data query - see Extended Data Mapping.

You may choose to calculate control limits and make them available to the end user to view with the other calculated statistics, while still plotting the saved control limits on the chart. The limits that are actually plotted on the chart follow an order of precedence. If limits are returned in the data query and are mapped into the template on the Ext Mapping tab, these have the highest priority. If you define a query to retrieve the limits, those are the second highest priority, and will be used on the chart. The third highest priority are the static values for Lower Control Limit, Centerline, and Upper Control Limit. The same order of precedence is used for the specification limits.

The Upper CL Source is the query template that returns the control limits to be used for the upper chart.

Similarly, the Lower CL Source is the query template that returns the control limits to be used for the lower chart.

If you specify a query, but at runtime the query returns no results, then the chart will use the static limits from this tab.

If you choose to specify a query, you may make it a parameterized query. The parameters may be set at runtime, as with any query. However, if they are static for a chart, you may type in the values on the Query Param entries on this tab. The default values for Param.1 and Param.2 for each query are [CHARTNAME] and [CHARTTYPE] respectively. By putting these values in square brackets [], the system recognizes them as keywords, and automatically substitutes chart parameter values that match the parameter names into the queries. For example, an iSPCChart template has both a ChartName and a ChartType parameter, found on the "General" tab. You define the chart name such as "SalesOrdersSPC," and the chart type is selected from a drop down list such as "XBAR." By using these parameters, you can write a parameterized query that returns control or specification limits for more than one chart or chart type. At runtime, the [CHARTNAME] and [CHARTTYPE] tokens are replaced with "SalesOrdersSPC" and "XBAR" and assigned to Param.1 and Param.2 in the associated CL Source template.

If you wish to establish control limits and store them within the system, you may do so. The default storage mechanism saves the control and specification limits by chart name and chart type, so a single query can be used for all system control limits.

The control limit queries and the specification limit query must always return data in the following format:

Column 1: Starting Effective Date

Column 2: Lower Control Limit / Lower Specification Limit

Column 3: Centerline Value / Target Value

Column 4: Upper Control Limit / Upper Specification Limit

Refer to the iSPCChartComponent Upper Chart and Specification Limit Parameters table for more information.

Alarms

The Western Electric Company (WECO) rules may be applied to the chart in any combination to evaluate control chart run alarms. You may also customize the default behavior of the alarm evaluation on a template-by-template basis. The WECO rules are applied on a subgroup level and do not indicate trends of points within a subgroup.

The alarms may be enabled individually for upper and lower charts as appropriate. The general setup for an alarm requires two integer values, the limit value and the length value. These two values are used in tandem. An alarm is considered active when x of y successive subgroup points have exceeded a threshold, where x = limit value, and y = length value. The default values are shown below.

For example, if 2 (limit value) of 3 (length value) subgroup points are beyond Zone A, the Zone A alarm rule has been violated. Zones A, B, and C represent the 3 sigma, 2 sigma, and 1 sigma widths beyond the centerline, respectively.

The Control Limit alarm rule is typically in violation if any point exceeds the control limit. The same is true for the Specification Limit alarm. Both their defaults are 1 point in violation.

The Alternating Limit alarm means x of y points have changed the side of the centerline that they are on.

The Run Limit alarm is activated when at least x of y consecutive points appear on the same side of the centerline (they may be above or below). The Trend Limit alarm is activated if x of y points indicate the variable is headed in the same direction (either increasing or decreasing).

The Zone A and Zone B alarms indicate that x of y points are in or beyond the indicated zone (e.g., 2 of 3 in Zone A or Beyond). There are two distinct alarms for Zone C: all x of y points inside the zone and all x of y points outside the zone.

The Sigma Different Limit alarm refers to an alarm which monitors the values of consecutive subgroup plot points. If they vary by 4 sigma, an alarm is generated.

Refer to the iSPCChartComponent Alarm Parameters table for more information.

Behaviors

This tab is used to configure which functions will be enabled or disabled when the user right-clicks the mouse on an iSPCChart object at runtime. For details on the individual parameters configured by this tab, refer to the iSPCChartComponent Behavior Parameters table.

Refresh Page

This page is used to control the automatic refresh behavior for the object. For details on the individual parameters configured by this page, refer to the General Applet Common Properties/Methods table.

Security

See the Security information in the Query Editor topic.

[iGrid Editor Reference](#)

iGrid Editor

This editor is used to create and edit display templates for the iGrid object. Much like the iChart object, the iGrid object is a fairly flexible applet, and as a result, has a large number of configurable parameters and behaviors.

General

This tab is used to configure general parameters and behaviors to the iGrid object, including the type of grid, fonts, and colors, as well as which elements of the grid are to be displayed at runtime. For details on the individual parameters configured by this tab, refer to the iGridComponent Properties tables.

Data Mapping

This tab allows you to pick explicit columns from a query result set to display in the grid. These are then used on the layout tab to further define the grid runtime characteristics. Columns returned by the query can be suppressed from the grid or reordered using this tab.

Layout

This tab is used to override column headings and widths in the grid, and to specify row headings for certain grid modes. The Display Columns are set on the Data Mapping tab (see above). You can specify specific columns headings, widths and formats for each column to be displayed (the column headings default to the returning data columns if none are set). For details on the individual parameters configured by this tab, refer to the iGridComponent Layout And Formatting Properties table.

Color Context

This tab is used to configure the context-sensitive color rules for the object. To edit a value, click the appropriate cell in the grid. The current column name, color, and values for the selected cell are displayed. To set or change one of the values, edit it in the displayed fields, and then select the associated Set button. Colors can be mapped on a per-row or per-column basis depending on the grid type. The Scoreboard and ColumnLights modes use per-column highlighting, whereas the other modes use per-row highlighting. For details on the individual parameters configured by this tab, refer to the iGridComponent Context-Sensitive Color Properties table.

You can also insert and delete rows and columns. Click the **Insert Column** button to insert a blank column at the location of the highlighted cell and shift the remaining columns one column to the right. Click the **Insert Row** button to insert a blank row at the location of the highlighted cell and shift the remaining rows down one row. Click the **Delete Column** button to remove the column at the location of the highlighted cell and shift the

remaining columns one column to the left. Click the **Delete Row** button to remove the row at the location of the highlighted cell and shift the remaining rows up one row. Use the **Match Mode** radio button to specify which mode, String or Mode, is used to match data values.

Behaviors

The right-click menu behavior is set by this tab. You can enable/disable right click functions through this interface.

Refresh Page

This tab is used to control the automatic refresh behavior for the object. For details on the individual parameters configured by this tab, refer to the General Applet Common Properties/Methods table.

Security

See the Security information in the Query Editor topic.

[iBrowser Editor Reference](#)

iBrowser Editor

This editor is used to configure the properties for the iBrowser object. The iBrowser object is a fairly simplistic object that can display lists and trees of selectable data. The color settings apply only to the "tree" mode.

General

This page is used to set the parameters that control the appearance and behavior of the iBrowser object. For details on the individual parameters configured by this page, refer to the iBrowser Properties table.

Data Mapping

Data Mapping allows you to select which columns from a returning query that you wish to display in the iBrowser, and in what order. By browsing to a query, you can move the columns you wish to see to the Display Columns list box (or enter the column name via the Column Name text box). You can reorder the columns by using the Up and Dn (Down) buttons.

Security

See the Security information in the Query Editor topic.

[iTicker Editor Reference](#)

iTicker Editor

This editor is used to create and edit display templates for the iTicker object.

General

This tab is used to configure general parameters and behaviors to the iTicker object. For details on the individual parameters configured by this tab, refer to the iTickerComponent Properties tables.

Data Mapping

This tab is used to select which columns from a query are to be displayed in the iTicker object (the default is all columns). The Browse button (...) can be of great value when using this tab, as it provides a means for returning a list of available columns for a particular Query Template. For details on the individual parameters configured by this tab, refer to the iTickerComponent Layout And Formatting Properties table.

Color Context

This tab is used to configure the context-sensitive color rules for the object. To edit a value, click on the appropriate cell in the grid. The current column name/color/values for the selected cell will be displayed. To set/change one of the values, edit it in the displayed fields, then select the associated "Set" button.

Additional buttons are provided to insert/delete rows and columns. The "Insert Column" button inserts a blank column at the location of the highlighted cell, shifting the remaining columns one column to the right.

The "Insert Row" button inserts a blank row at the location of the highlighted cell, shifting the remaining rows down one. The "Delete Column" button removes the column at the location of the highlighted cell, shifting the remaining columns one column to the left. The "Delete Row" button removes the row at the location of the highlighted cell, shifting the remaining rows up one row. The Match Mode radio button specifies which mode will be used to match data values, String or Number. Colors can be mapped on a per-row or per-column basis, depending on the value of the ColorByColumn flag. For details on the individual parameters configured by this tab, refer to the iTickerComponent Context-Sensitive Color Properties table.

Refresh Page

This tab is used to control the automatic refresh behavior for the object. For details on the individual parameters configured by this tab, refer to the General Applet Common Properties/Methods table.

Security

See the Security information in the Query Editor topic.

HTML Development Add-In

FrontPage 2000 (part of Microsoft Office 2000), FrontPage 2002 (part of Microsoft Office XP), FrontPage 2003 (part of Microsoft Office 2003), and/or Dreamweaver MX (from Macromedia) can be used to generate Web pages to be used to access data from an SAP xApp Manufacturing Integration and Intelligence (SAP xMII) server. There is an SAP xMII add-in that can be installed and utilized to make this process even easier. For a quick tour through the process, refer to the Quick Start Guide to Generate an Output Page with FrontPage. This guide will go into more details on all the settings available in the Add-In.

The Script Assistant helps you develop scripts within the Web page so that you can link user actions to page functions and assists you in linking SAP xMII applets to each other and to user interaction events. See the Script Assistant User's Guide for more information.

Installation

Refer to the FrontPage Add-In Installation Guide or to the Dreamweaver Add-In Installation Guide.

Using the Add-in

Similar to the Page Generator, you can select the query and the display templates to use to generate HTML. Unlike the page generator, using FrontPage you can insert any combination of display elements that you wish into a single Web page.

The SAP xMII Object tool is divided into the following areas:

Button Bar

Query Templates

Display Templates

Object Selection

Dimensions

Each of these sections is described below.

Button Bar

The button bar is composed of six buttons.

The Object Mode allows the user to select a previously created and saved object and insert that object into the page.

The Refresh Lists button repopulates the Query Templates and Display Templates windows to reflect any changes.

Create HTML inserts the defined object into the current page being edited in FrontPage.

The Save As Object button allows the user to create a complete display element (Query + Display + Parameters) and save it for reuse.

The Login button can be used to log onto the SAP xMII server.

The Close Wizard button allows you to return to the FrontPage editor without losing the current settings and selection in the SAP xMII Object tool. Simply re-select the SAP xMII Object choice from the Tools menu, and the window will re-appear.

Query Templates

The Query Templates section is divided into three elements. The Query Type buttons to the right of the Query Template selection window allow you to filter the selection to the chosen query type. The selection windows allow the user to select the appropriate folder and query to be used in the creation of the graphical object.

Display Templates

The Display Templates section is also divided into three elements. The Display Type buttons to the right of the Display Templates allow you to filter the available selections to the chosen chart type. The selection windows allow the user to select the appropriate folder and display element to be used in the creation of the graphical object.

Object Selection

The Object Selection area is divided into three areas, the Object Type selection, the Object Details, and Parameters. By selecting the object type that you wish to insert (Applet, Graphic, Hyperlink, or Report), different object details will be available to set. Similarly, the parameters relevant to the selected object type are available from a list box. All the parameters may be edited after the object has been inserted by right clicking on the object, and selecting Java Applet Properties. These selections are described further below.

Object Type

Applet

If you select an applet for insertion to the page, you will be able to enter the applet's reference name. Many of the parameters of the display applets are exposed to scripting, so that you can change settings based on preference, or even have a second applet change its data query based on the data selection in another display object.

For example, you could insert a grid object with a list of the week's orders or batches. You could then insert a trend or other object that displays critical parameters for the order or batch chosen in the grid object. The layout and interaction possibilities are up to you. The Applet Reference Guide discusses each applet's set of parameters in detail. For an introduction to the SAP xMII Applets, see the Applet User's Introduction in the xMII help documentation.

Graphic

The image type is the only detail to be set for the graphic object. Only a chart object may be returned as a graphic object. This setting produces a chart returned to the page as a GIF, JPEG, BMP or WBMP formatted graphic.

Hyperlink

If you select to insert a hyperlink, you will be asked to enter the output format of the data, and the name of the hyperlink. You may only select to return chart data to a hyperlink request. This will insert a hyperlink onto the Web page. When that hyperlink is pressed, the current window will be replaced by the data in the chosen format.

If the return format is HTML, you may also set an automatic refresh rate and apply a stylesheet. When selected, the hyperlink returns the data in the chosen format to the requester. Selecting an embedded GIF return format as a graphic is similar to asking for an HTML return, except that you cannot format the chart legend and tag details.

If you choose to return XML, you may also apply a XSL stylesheet to the XML document.

If you choose to return WML, the results can be displayed on WAP-enabled wireless phone.

Finally, choosing a Comma Separated Values (CSV) output file will allow you to save the returned data to a local file for further analysis.

Report

This is for use on SAP xMII Report pages (.irpt). Typically the /Illuminator/StyleSheets/IllumRowsetTable.xsl stylesheet is used to display the QueryTemplate output as an HTML table on the report page. To choose an XSL stylesheet, either type the fully qualified path to the stylesheet or double click on the stylesheet input field to open the Select Stylesheet dialog.

Parameters

Parameters may be selected one at a time via a list box, or you can type the name value pairs manually. To access the list box, click the Browse button next to the Parameter Name text box. A list of user-defined HTML localization tokens that can be used to localize your HTML appears. You must explicitly add each name value pair to the parameter list box to override the defaults. For a detailed explanation of parameters, refer to the Applet Reference Guide.

Dimensions

The dimensions text boxes allow the user to set the width and height of the object in pixels.

HTML Localization Assistance

Using keys to localize your HTML is explained in the localization guides. To get a list of the keys that you can use, press the browse button next to the Parameter Value: text box on the tool. You will get a popup with a list of all the keys available on the server.

Select one, press OK, and the correct syntax for string replacement will be inserted into the Parameter Value text box {##TokenName}. You can use the key as a parameter value, or simply cut and paste it anywhere within your HTML page.

[Using FrontPage to Generate an Output Page](#)

After you establish a connection to a database and develop query and display templates, you can utilize our Microsoft® FrontPage® add-in. For advanced documentation on the SAP xApp Manufacturing Integration and Intelligence (SAP xMII) Wizard refer to the HTML Add-In User's Guide.

Open FrontPage. Select *File --> New --> Web*. A *Web* is a collection of common Web files managed by FrontPage. If you already have a Web site configured, you can add a new page to it. You must open a *Web* on the SAP xMII Web server to view the query and display templates.

Choose *One Page Web* and select the save location. The save location should be on or below the Web root directory of your Web server. Hit *OK*. By default, a new Web page will open. Now we can use the *SAP xMII Wizard* to insert the HTML code for the applet.

But first we must install it. Go to *Tools --> Add-Ins*.

Browse to *C:\Lighthammer\Illuminator\Wizards*.

Select the *Illum10FPWizard.dll*. Hit *OK*.

Click on the check-box next to the *SAP xMII Wizard*. Hit *OK*.

Select *Tools --> Illuminator Object*.

The SAP xMII Wizard will load.

Use the windows on the left to browse to the locations of the query template and the display template we created in the Quick Start - Templates section. You can use the buttons on the right to filter query and display types. The bottom section allows you to add parameters to overload the default characteristics of the templates. Use the *Create HTML* button to insert the applet code into our HTML page. (The Add-In is explained in much greater detail in the HTML Add-In User's Guide.)

Now we must save the page into our Web. Hit *File --> Save As*. Give the chart a name and hit *Save*. Remember to save the page to a location at or below your Web server's root directory.

Finally we can load the page using a Web browser. It is important to address the page using the Web server and not the local file system. For the applet to pull data you must use the HTTP protocol.

Script Assistant User Guide

Script Assistant

The Script Assistant improves scripting syntax accuracy and reinforces lessons on the SAP xMII Applet Object Model. You are provided with an exploration experience of events, method calls, and properties. The tool interactively creates stub function calls based on user selections.

Benefits

Takes the guesswork out of the object's coding syntax and spelling

Inserts correctly formatted Property Name-Value pairs into the Applet definition blocks

Inserts stubbed-out function calls linking one applet to another based on the selected applet event, jump-starting the page development cycle

The Script Assistant can be launched from Microsoft® FrontPage® once the FrontPage add-in has been properly installed. The FrontPage plug-in works in conjunction with previously created .html or .irpt pages by extracting existing page applet definitions and populating the tool with appropriate events and methods. In contrast, the standalone version retrieves default applet definitions, helping the user create syntactically correct code that can be copied and pasted into any HTML editor.

If you created two SAP xMII applets on a page and call the Script Assistant from the Tools menu, you have the ability to script a single applet or link one applet event to another applet. The Target Applet drop-down list allows you to select any applet on the current page. You can then select the desired event of the target applet from the Target Event drop-down list. The Function Name is used to create the actual JavaScript function and is linked to the target applet event that you chose.

You can then select the applet to which you want to link. It can be the same applet as the target or it can be any other applet on the page. For example, you want to create a JavaScript function named `BatchList_Selected`. In the example, the target applet is the `BatchList`. On a selection event from the `BatchList` applet, you want to change a parameter in the `BatchTrend` applet so the currently selected batch in the list is the one shown in the trend applet. In the Applet drop-down list, choose the `BatchTrend` applet. In the Applet Method drop-down list, select `getQueryObject()` because you want to set a query parameter. Choose `setParam` method to set `Param.1` to a specific value (the batch number of the selected row in the `BatchList` applet) and update the trend.

Press the Insert key to insert a specific `applet.object.method` into the text area shown. You can continue to select different applets, objects, and methods from the Applet, Applet Method, and Object Method drop-down lists, and insert syntactically correct script. You can also manually edit the script in the text area or create it on the page and edit it within the HTML view.

When you press the Save Script button, it creates a function (a function named `BatchList_Selected`), inputs a parameter into the `BatchList` applet definition linking the selection event to the `BatchList_Selected` function, and inserts the JavaScript text into the script block.

The standalone version displays only default applet definitions and is not capable of referencing specific applets in the targeted drop-down list boxes. Its primary use is a quick lookup of the proper syntax for applet methods, as shown below. You can input as many methods as you like, and when you click the Save Script button, it saves the text in the text area to the clipboard from which you can then paste the JavaScript text into any HTML editor.

Script Assistant User Control Definitions

Control Name	Description	Tool Usage	Notes
Target Applet	Host applet name	FrontPage only	Lists all of the applet names that are found on the open .html or .irpt page
Target Event	Host applet event	FrontPage only	Lists all of the supported events for the applet selected in the Target Applet control
Function Name	Host applet function	FrontPage only	Displays the function name that is created by the Script Assistant after the Target Event is selected. It is the name of the function that is inserted into the .html or .irpt page by the tool.
Applet	Applet name	FrontPage or standalone	When used in FrontPage, this control lists only the applet names found in the open Web page. When run as a standalone application, this control lists all SAP xMII applet types.
Applet Method	Applet method's name	FrontPage or Standalone	Lists all of the base applet methods that apply to the type of applet selected in the Applet control. This method is inserted into the text area by the tool when you click the Insert button.
Object	Object	FrontPage	Lists the object method names that apply to the

Method	method's name	or standalone	selected applet method. It is only possible if the object method is supported by the applet. The completed expression is inserted into the text area by the tool when you click the Insert button.
Insert	Insert button	FrontPage or standalone	Inserts the selected applet method call into the text area immediately after the cursor
Include Document Object Reference	Auto-prefix document object reference	FrontPage or standalone	If this parameter is selected, the tool adds the document object as a prefix to the applet name. Choose this option before clicking on the Insert button. It is used to properly reference an applet or a control on the page. We recommend that the document object use a prefix whenever possible. For example, an applet method is typically in the following format: document.BatchTrend.getQueryObject(); where the "document." is the document object prefix.
Save Script	Save Script button	FrontPage or standalone	Copies the text in the editor to the paste buffer if the tool is standalone. In the case of the FrontPage plugin, the text in the editor box is inserted into the corresponding function name and the target applet has a new parameter name-value added referencing the new script function.
Delete Script	Delete Script button	FrontPage only	Removes the script block and applet's function reference defined by the combination of the selected items found in the Target Applet and Target Event controls.

Script Assistant Example

To illustrate how the Script Assistant might be used, consider an .html page that contains the familiar BatchTrend chart and the BatchList applets. The applet definition for the BatchTrend applet is as follows:

```
<APPLET NAME="BatchTrend" WIDTH="715" HEIGHT="45" CODE="iGrid"
CODEBASE="/Illuminator/Classes" ARCHIVE="illum8.zip" MAYSCRIPT>
<PARAM NAME="QueryTemplate" VALUE="Training/BatchTrendQuery">
<PARAM NAME="DisplayTemplate" VALUE="Training/BatchTrendChart">
</APPLET>
```

and the applet definition for the BatchList applet is as follows:

```
<APPLET NAME="BatchList" WIDTH="715" HEIGHT="45" CODE="iGrid"
CODEBASE="/Illuminator/Classes" ARCHIVE="illum8.zip" MAYSCRIPT>
<PARAM NAME="QueryTemplate" VALUE="Training/BatchListQuery">
<PARAM NAME="DisplayTemplate" VALUE="Training/BatchListGrid">
</APPLET>
```

Assume you launch the Script Assistant and make the necessary selections. Select the **BatchList** applet as a host target. It uses *SelectionEvent* to call the *BatchList_Selected* function. Choose the **BatchTrend** applet, the *getQueryObject()* object, and a corresponding *setParam(Index, Value)* object method. When you click the **Insert** button, the *document.BatchTrend.getQueryObject().setParam(INDEX, NEWVALUE);* expression is inserted into the editor's text box. When you click the **Save Script** button, the following changes are made to the applet and the JavaScript.

A new function is created. Note that the function is placed in the head section of the Web page, inside its own script block with a unique ID.

```
<SCRIPT Language="JavaScript" id="BatchList_Updated">
```

```
function BatchList_Selected() {  
document.BatchTrend.getQueryObject().setParam(INDEX, NEWVALUE);  
}  
</SCRIPT>
```

The BatchList applet is changed to include an additional parameter name-value pair with Name=SelectionEvent, shown below immediately before the </Applet> tag:

```
<APPLET NAME="BatchList" WIDTH="715" HEIGHT="45" CODE="iGrid"  
CODEBASE="/Illuminator/Classes" ARCHIVE="illum8.zip" MAYSCRIPT>  
<PARAM NAME="QueryTemplate" VALUE="Training/BatchListQuery">  
<PARAM NAME="DisplayTemplate" VALUE="Training/BatchListGrid">  
<PARAM NAME="SelectionEvent" VALUE="BatchList_Selected">  
</APPLET>
```

This is a simple example and needs additional logic to be functional; however, it shows the power of the Script Assistant when it is used within FrontPage. Many typographical errors can be automatically eliminated. There is no need to look up specific syntax. This tool helps the developer properly code to the SAP xMII object model. While it does not develop the complete code, it does make the syntax for methods, events, and function calls as easy as point and click. The productivity of the Web page developer can be greatly enhanced by taking advantage of this tool.

Query Error Handling Example

You can use the following object methods to provide query status:

isDataValid(); is a validity flag that shows "true" or "false." For example,
document.appletName.getQueryObject().isDataValid();

getLastStatusCode(); shows a numeric status code. For example,
document.appletName.getQueryObject().getLastStatusCode();

getStatusMessage(); shows the last message. That is
document.appletName.getQueryObject().getStatusMessage();

For a detailed description of the applets, please see the Applet General Reference and the Applet Object Model Reference. Also, you can refer to the Web Scripting with SAP xMII Objects topic in the xMII Help documentation.

Standalone Wizards

Similar to how you can generate HTML directly into FrontPage or Dreamweaver, you can generate HTML to be placed into any editor. For a detailed description of the various buttons and controls on the wizard, refer to the HTML Add-In help. This section highlights the usage differences between the stand-alone wizard and the FrontPage or Dreamweaver add-in.

You can also use the Script Assistant in stand-alone mode. See the Script Assistant User's Guide for more information.

Installation

The stand-alone executables are installed on the Web server during the installation process. On computers other than the Web server, a link to a Web package installation can be found on the main menu under Visualization Services -> Wizard Install.

Using the Stand-alone Wizard

When you start the wizard, it prompts you for the name or IP address of the Web server (defaults to 'localhost'). It does this so that it can query templates and save objects to and from the appropriate Web server location. You must provide the correct server name or IP address as well as the necessary user credentials.

When you press the Create HTML button, the HTML is copied onto the Windows Clipboard. You can then paste it directly into any HTML editor that you are using to create your Web pages.

Using the Stand-alone Script Assistant

In stand-alone mode, the script assistant has no access to the html page and any applets that might exist. As a result of this the Target Applet, Target Event, and Function Name controls are all disabled, and the Applet control does not use actual applet names, but provides an example of each type.

When you press the Save Script button, the text located in the script area is copied onto the Windows Clipboard. You can then paste it directly into any HTML editor that you are using to create your Web pages.

Dynamic Page Generator

The main menu includes a link to the Dynamic Page Generator that facilitates automatic generation of HTML page content or a ChartServlet hyperlink/URL. This is designed to exploit the template capabilities of the system to provide an interactive means for creating HTML code by simply selecting a query template and a display template and specifying the dimensions of the desired applet.

Selecting either of the two buttons will open a new browser window and create the appropriate Web content automatically. In the case of the page generator, you can use the "View Source" option of your browser to view, select, and cut/paste the HTML code. In the case of the GIF chart generator, you can copy the URL and create a hyperlink in another Web page to trigger the generation of the chart when selected.

Related Content

Please refer back to the full xMII help documentation which can be found at:

http://help.sap.com/content/documentation/xapps/docu_xapps_mii.htm

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.